

Advances in Computational Syntax and Semantics

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Jesús Figueroa (Mexico)
Alexander Gelbukh (Russia)
Ioannis Kakadiaris (USA)
Serguei Levachkine (Russia)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

María Fernanda Ríos Zacarias

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 117**, septiembre de 2016. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 117**, September 2016. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 117

Advances in Computational Syntax and Semantics

Alexander Gelbukh (ed.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2016

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in DBLP, LATINDEX, and Periodica
Indexada en DBLP, LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

Computational syntax and computational semantics have been at the core of natural language processing technology since its inception in 1960s, and have always been considered a classical research field in computational linguistics. However, these fields continue to be actively developed. While computational syntax can be considered a well-understood area, computational semantics is still to find its way towards reconciling expressive power with technically affordable ways of text analysis.

Recent decades have seen a strong trend of moving away from rich text representations, in favor of simple, or even simplistic, modeling of texts as feature vectors. Such simplistic representations have two advantages: on the one hand, they are relatively simple to obtain from real-life texts with all their inconsistencies and grammatical peculiarities, and, on the other hand, they permit very complex and efficient processing algorithms, specifically, machine learning-based classifiers ranging from support vector machines to modern-day deep neural networks. Generalizing such machine-learning algorithms to text representations richer than feature vectors is non-trivial.

However, a disadvantage of purely statistical machine-learning methods based on feature-vector text representation is their inability to support complex reasoning, which has been long considered the basis of human language understanding and a necessary pre-requisite of any program aimed to flexible and reasonably complete natural language understanding.

It is thus on the agenda of natural language processing research now includes the search of ways of combining the advantages of the two worlds: the flexibility and efficiency of machine-learning methods and the power of logical reasoning over structured text representations. I see two ways to do it: first, extracting more complex features, such as recently introduced syntactic n-grams, from structured semantic representations, and feeding these features into machine-learning algorithms; secondly, and more interestingly, using machine-learning algorithms for decision-making in graph- and network-based logical reasoning algorithms.

This justifies a closer look at syntax and semantics of natural language, as well as on their applications in various tasks of natural language processing, such as question answering and machine translation.

An important and still unresolved issue in natural language processing technologies is the addressing of languages outside the major European languages. This includes attention to languages typologically quite different from Germanic and Romance languages, which have received most of attention in computational linguistics. This also includes looking at European languages from the perspective of a grammar tradition outside of the Greek-Latin mainstream. Accordingly, I selected for this issue some papers devoted to Arabic and two Turkic languages, as well as one paper devoted to the description of English grammar within a Sanskrit-based linguistic tradition.

The papers included in this special issue were selected basing on a double-blind review procedure, with participation of 126 leading experts in the topic from 38

countries. A list of international reviewers that participated in the selection process is given at the end of this volume.

This special issue will be useful to researchers, students, and engineers working in natural language processing and computational linguistics, especially in the subfields of computational syntax and computational semantics as well as their practical applications.

November 2016

Alexander Gelbukh,
Guest Editor

Table of Contents

	Page
Analyzing English Phrases from Pāṇinian Perspective	9
<i>Akshar Bharati, Sukhada, Dipti M Sharma</i>	
A Modified Earley Parser for Huge Natural Language Grammars	23
<i>Sinan Polat, Merve Selcuk-Simsek, Ilyas Cicekli</i>	
Extracting HPSG Lexicon from Arabic VerbNet	37
<i>Noureddine Loukil, Kais Haddar</i>	
A Survey of Arabic Text Representation and Classification Methods	51
<i>Rami Ayadi, Mohsen Maraoui, Mounir Zrigui</i>	
A Word Embeddings Model for Sentence Similarity	63
<i>Victor Mijangos, Gerardo Sierra, Abel Herrera</i>	
Using Multiple Metrics in Automatically Building Turkish Paraphrase Corpus	75
<i>Bahar Karaođlan, Tarık Kışla, Senem Kumova Metin, Ufuk Hürriyetođlu, Katira Soleymanzadeh</i>	
An Approach to Content Extraction from Scientific Articles using Case-Based Reasoning	85
<i>Rajendra Prasath, Pinar Öztürk</i>	
Event Causality Extraction from Natural Science Literature	97
<i>Biswanath Barik, Erwin Marsi, Pinar Öztürk</i>	
Discourse Connective - A Marker for Identifying Featured Articles in Biological Wikipedia	109
<i>Sindhuja Gopalan, Paolo Rosso, Sobha Lalitha Devi</i>	
Named Entity Recognition for the Agricultural Domain	121
<i>Malarkodi C. S., Elisabeth Lex, Sobha Lalitha Devi</i>	
Question Answering Based on Temporal Inference	133
<i>Zeineb Neji, Marieme Ellouze, Lamia Hadrach Belguith</i>	

Statistical Approach for Spontaneous Arabic Speech Understanding Based on Stochastic Speech Recognition Module	143
<i>Aymen Trigui, Naim Terbeh, Mohsen Maraoui, Mounir Zrigui</i>	
Initial Experiments on Russian to Kazakh SMT	153
<i>Bagdat Myrzakhmetov, Aibek Makazhanov</i>	
Contribution to the Achievement of a Spellchecker for Arabic.....	161
<i>Khaireddine Bacha, Mounir Zrigui</i>	
Emerging Technology Enabling Dyslexia Users To Read and Perceive Written Text Correctly.....	173
<i>Tereza Pařilová, Filip Mrváň, Bruno Miřík, Eva Hladká</i>	

Analyzing English Phrases from Pāṇinian Perspective

Akshar Bharati, Sukhada, and Dipti M Sharma

Language Technologies Research Centre,
IIIT Hyderabad, Telangana 500032, India
sukhada@research.iiit.ac.in
dipti@iiit.ac.in

Abstract. This paper explores Pāṇinian Grammar (PG) as an information processing device in terms of ‘how’, ‘how much’ and ‘where’ languages encode information. PG is based on a morphologically rich language, Sanskrit. We apply PG on English and see how the Pāṇinian perspective would deal with it from the information theoretical point of view and its effectiveness in machine translation.

We analyze English phrases defining *sup* (nominal inflections) and *tin* (finite verb inflections) and compare them with the notion of *pada* (an inflected word form) and *samasta-pada* (compound) in Sanskrit.

Sanskrit encodes relations between nouns and adjectives and nouns in apposition through agreement between gender, number and case markers, whereas English encodes them through positions. As a result, constituents are formed. It appears that an English phrase contains more than one *pada*, hence, cannot be similar to a *pada*. However, we show the linguistic similarities between a *pada*, *samasta-pada* and ‘phrase’.

1 Introduction

Languages encode linguistic information in terms of explicit markers or positions of the words. This inspires us to analyse how a source language syntax encodes linguistic information, so that it can be transferred to target language. Pāṇinian Grammar (PG) gives insights to explore ‘how’, ‘how much’ and ‘where’ languages encode linguistic information [6]. In this paper, we use concepts from PG and apply them to English and show how it can help in machine translation (MT).

PG analyses a word as a combination of ‘root’ (*prakṛti*) and ‘suffix’ (*pratyaya*) [9]. Pāṇini uses the term *pada* for the words that are ready to participate in a sentence. A *pada* contains explicit information about a word’s semantic relation with other words in a sentence.

The word derivation process in PG takes a nominal stem and/or a verbal stem as the basic input and terminates the process with the derivation of *pada* [22]. Since a *pada* is formed with respect to an actual sentence structure, it is called the highest derivative and is a syntactic unit rather than a morphological unit in the Pāṇinian system [22].

Sanskrit uses two different terms, *śabda* and *pada*, both of which are roughly translated as ‘word’ in English. In Sanskrit, *śabda* is used for linguistic expressions ranging from an individual speech sound to an utterance [23], whereas, a *pada* is a primary syntactic unit that appears in a sentence. In other words, the difference between a *śabda* and a *pada* is that of a ‘word’ and its ‘fully inflected forms’ [18]. A *śabda*/word is a language unit such as lexeme, word or word form whereas a *pada* is a word form that has inflected to mark its semantic relation with other words. Pāṇini categorizes a *pada* into two classes: 1) *subanta* and 2) *tinanta* [21, 8].

1. *Subanta*: A *subanta* class includes the participants of a sentence which inflect for marking various semantic relations. A *subanta* is formed by suffixation of nominal inflections called *sup*¹.

The *subanta* class includes all nouns, pronouns, adjectives and adverbs etc. in it. Since, adverbs are indeclinables, they do not inflect for any case. That is why adverbs do not seem to carry any *sup* on surface. The sūtra (A.2.4.84) deletes the inflections attached to the adverbs. It suggests that at some point of time, adverbs also had been inflecting like other nominals. Hence, adverbs also fall in the *subanta* class.

2. *Tinanta*: This class includes words which mark some semantic relations and finiteness of the verb. In Sanskrit, verbs take *tin*² suffixes to express tense, number, person, mood and voice. Similarly, in English, the auxiliaries and modal verbs when attached to a verb express tense, number, etc. hence correspond to *tin*.

The primary objective of analyzing a sentence is to identify what role each part is playing. The role of the finite verb becomes important with relation to other participants. Therefore, Pāṇini takes only two classes.

Though this classification is mainly based on the surface realization of the words in Sanskrit, Pāṇini’s concepts of grammar are not specific to Sanskrit. The concepts are rather generic and can be applied to other languages. The major theoretical concepts from PG would directly apply on other agglutinative languages, languages similar to Sanskrit. However, they can be extended to other languages as well.

In Sanskrit, the nominal inflections *sup* and the finite verb inflections *tin* are realized through suffixation. However, the syntactic mechanisms for marking the semantic relations across words in a sentence might be different in different languages. Some languages such as Persian might have prepositions, some might have other kind of syntactic devices. For example, in Hindi, the relations of a noun to the verb or other nouns are marked through postpositions. From the Pāṇinian perspective, a *pada* in Hindi would be ‘noun+postposition’ [5].

¹ *Sup* is the acronym formed from the first and the last phoneme of the list of nominal suffixes.

² *Tin* is the acronym formed from the first and the last phoneme of the list of verbal suffixes.

Sanskrit has a grammatical rule *apadaṁ na prayuñjīta* [11] which says: “a word which is not a *pada* should not be used in a sentence” [8]. Sobin’s statement that “only phrases may be sentence fragments” [24] imposes a constraint similar to the statement *apadaṁ na prayuñjīta* for English. It is alluring to compare the two concepts *pada* of Sanskrit and ‘phrase’ of English.

Application of PG to other languages and finding out its effectiveness for machine translation (MT) is the task on hand. In this paper, we look at English from the Pāṇinian perspective. We investigate equivalent mechanisms of *sup*, *tin* and *pada* in English phrases and compare them with the notions of *pada* and *samasta-pada* (compound) in Sanskrit.

We talk about the related work in Section 2. Section 3 describes the necessary conditions for *pada* formation according to PG. Section 4 defines *sup*, *tin* for English and compares English phrases with the notions of *pada* and *samasta-pada* in Sanskrit. Section 5 shows a continuum between phrases and compounds. Section 6 shows how complex phrases are handled using the Pāṇinian perspective. Section 7 concludes the paper.

2 Related Work

Gangopadhyaya has analyzed noun phrases in Bengali and studied assignment of role and the *kāraka* theory following the Indian grammatical tradition. According to her: “The term phrase corresponds to the term *pada* in its minimal form but not in its expanded form, i.e. when a phrase is understood as a syntactic constituent consisting of more than one word.” According to her, a single word phrase corresponds to a *pada* but a phrase that consists of more than one word does not correspond to a *pada* [12].

Gangopadhyaya does not account for phrases which consist of multiple words such as “brave soldiers”, “very intelligent boy”, etc.

According to Apte [2], the expression of a single idea is a word (*pada*) and the aggregation of two or more words without a subject or predicate is a phrase (*padasamuccaya*).

Apte calls a phrase as *padasamuccaya*, but does not give any linguistic account for it. He looks at Sanskrit from the English perspective, and therefore interprets a phrase as a group of multiple *padas*. But if we look at English from the Pāṇinian perspective, we find that a minimal/simple phrase corresponds to a *pada* or a *samasta-pada*. And, a complex phrase that is composed of two or more phrases corresponds to a *padasamuccaya*.

Local Word Grouping (LWG) [5] is a notion similar to *pada* found in literature. In LWG, word groups are formed on the basis of local (adjacent) word information for Indian languages.

Our assessment is based on flow of information where word groups are formed on the basis of neighboring syntactic inflections called *sup* and *tin* for any language. This is grammatically more precise and also allows to find out syntactic elements that unite the words of a sentence into a meaningful unit.

To the best of our knowledge, we have not found any work that analyzes English phrases from the Pāṇinian perspective.

3 *Samartha* Theory of Pāṇini and its Relation to *Pada* and Phrase Formation

Pāṇini's *samartha* theory stands as a fundamental principle for any semantic and syntactic operation. According to Pāṇini, no grammatical operation can take place, be it *pada* formation or sentence formation, until and unless they qualify the condition of being *samartha* [21]. Thus the concept of *sāmarthya*³ is a fundamental principle for any grammatical operation in a language string. The word *samartha* is used in the following two meanings:

1. *Ekārthābhāva sāmarthya*: It says that 'formation of a *pada* depends on unity of meaning' [26]. In this case, the *padas* having direct semantic relation become one *pada* as in compounds and primary and secondary derivatives. Here the word *samartha* means "organized together" (*saṃgatārtham samartham*) and "fused together" (*saṃsṛṣṭārtham samartham*) [17]. The objective of *ekārthābhāva sāmarthya* is to present compounds as one *pada* (*ekapada*) or one unit.
2. *Vyapekṣā sāmarthya* (meaning-interdependence): It says that "any operation pertaining to *padas* takes place if and only if the *padas* have direct semantic connection" [26]. In this case, *samartha* means "seen together" (*sampreksitārthaḥ samarthaḥ*) and "bound together" (*sambaddhārthaḥ samarthaḥ*) [17]. For example, subject, verb, object etc. are seen bound together in a sentence. The objective of *vyapekṣā sāmarthya* is to show sentence as one unit. *Padas* seem to carry diverse meanings but a sentence indicates a single meaning.

For a word to stand in a syntactic structure, it is necessary to pass through one of these *sāmarthyas*.

Let us take the Sanskrit sentence (1) and examine how Pāṇini captures the flow of information through his grammar.

- (1) Vīrāḥ sainikāḥ deśam rakṣanti
brave.PL,NOM soldier.PL,NOM country.SG,ACC protect.PR,3,PL
'Brave soldiers protect the country.'

In (1), the word *rakṣanti* is a *tiṅanta pada*. It is composed of the verbal base *rakṣ* and a *tiṅ* inflection namely *-anti*. A *tiṅ* inflection is assigned to a verb with respect to its compatibility (*sāmarthya*) with the doer/agent or theme/patient of the action. In the active voice, the *tiṅ* suffixes express the doer/agent of the action through agreement. The suffix *-anti* denotes active voice, third person, plural. When the doer is expressed by a *tiṅ* suffix, the sūtra (A. 2.3.46) [27] assigns nominative case (*prathamā vibhakti*) to the doer to express nominal stem

³ The words *samartha* and *sāmarthya* are used interchangeably in Sanskrit grammar.

meaning (*prātipadikārtha*), gender (*liṅga*), or number (*vacana*) etc., of the doer. This also makes the nominal a *subanta pada*.

In (1), the words *sainikāḥ* and *vīrāḥ* are marked with nominative case (*prathamā vibhakti*) and plural number. The agreement between the verb *rakṣanti* and nominal *sainikāḥ* (soldiers) indicates that the *pada sainikāḥ* plays the role of the doer/agent of the action *rakṣanti* (protect).

Having the same *vibhakti*, the words *sainikāḥ* and *vīrāḥ* express the modified and modifier relation between them and also confirm semantic compatibility (*sāmarthyā*) among them.

The theme/patient *deśam* (country) is marked with the *sup*, *-am* (accusative case, singular number). It makes *deśam* (country) a *subanta pada*.

From the above description, it is clear that a *pada* is a syntactic unit that takes a nominal or a verbal inflection called *vibhakti*, which explicitly marks the semantic relation of a word with another participant. In some cases, the *vibhakti* can also be NULL (zero) but it has to be present.

3.1 *Sāmarthyā* and Phrase Formation

While looking at *samartha* theory and the concept of phrase in English, we observed that both the theories capture the same aspect of language but in slightly different ways. They both capture the coherence of words together in a well formed syntactic structure. As per English grammar, a phrase is a sequence of words or sometimes a single word that functions as a single unit within a sentence [19, 16]. The words that are closely related to each other form a syntactic constituent. As per *samartha* theory, for a word to become a *pada* or to form a word group with other word/s such as in compounds, it has to have direct semantic relation with the other words in the sentence. This relates to the semantic relation among *padas* in a sentence and between the members of a compound.

According to the sūtra *samarthaḥ padavidhiḥ* (A. 2.1.1), any operation concerning to inflected word form/s (*pada/s*) i.e. *padavidhi* takes place if and only if there exists a semantic connection. So is true for constituent formation. Words in a sentence do not occur in isolation rather they form groups on the basis of their relationship with other words. Thus, the notion of *pada*, though developed for Sanskrit which has a rich inflectional and derivational morphology, can be applied to any language. If we apply it to English which is morphologically not so rich, we have to analyze English sentences from the Pāṇinian perspective, especially in terms of *sup* and *tiṅ* and *pada* and *samasta-pada* or *ekārthābhāva sāmarthyā* (single integrated meaning) and *vyapekṣā sāmarthyā* (meaning-interdependence).

4 *Subanta Pada* and *Tiṅanta Pada* in English

In Sanskrit, *sup* carries information of number and the case marker. However, English has different sets of morphemes for marking number and case information. In English, the number information is marked through a suffix ‘-s’ for

plural and ‘-0’ (NULL) for singular and the case is realized through prepositions or through ‘generalized *vibhakti*’ [3, 4] in terms of the position of the subject or object [1]. For example, in “to the boy” and “to the boys”, the preposition “to” marks case information and the inflection “-s” marks number information.

In English, the *tiñ* inflections are realized through auxiliaries and modal verbs. The *tiñ* suffixes also express the role of one of its participants through agreement.

Attachment of a nominal or a verbal inflection to a nominal or verbal entity makes it a *subanta* or a *tiñanta pada* respectively.

Let us take the English sentences in (2) and analyse it using Pāṇinian primitives such as *sup*, *tiñ* and *pada*.

(2) She gave books to Mohan.

Figure 1 shows the constituency tree diagram for sentence (2). The prepositional phrase “to Mohan” carries preposition “to” as a *sup*, hence, it corresponds to a *subanta pada*. The verb *gave* has *-ed* as a *tiñ* suffix, hence it can be considered a *tiñanta pada*.

The words “She” and “books” are the subject and object respectively and do not seem to have any explicit case marker. Bharati et al (1996,1998) have shown that English has the notion of ‘generalized *vibhakti*’ which corresponds to the *sup* suffixes in Sanskrit. The ‘generalized *vibhakti*’ is realized either through subject⁴ or object positions or through prepositions. Thus in sentence (2), *She* occurring at the subject position seems to carry no *sup*, but according to Bharati et al. (1996,1998), since it occurs at the subject position, it carries a generalized *vibhakti* in terms of subject position, hence, it is a *subanta pada*. Similarly, the object “books” carries a generalized *vibhakti* in terms of object position, hence, it is also a *subanta pada*. See Figure 1, where each box represents an independent *pada*.

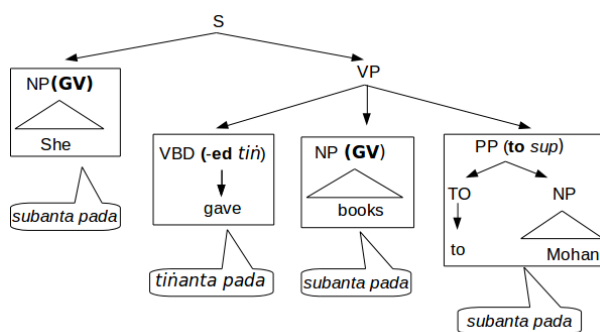


Fig. 1. Pada information in tree diagram for (2). In this figure GV stands for ‘generalized *vibhakti*’

⁴ In linguistics, the notion of subject in ILs is much debatable [7].

In sentence (2) each phrase consists of a single lexical item, therefore, we can say that a phrase with a single lexical item corresponds to a *pada* from the Pāṇinian perspective. But a phrase is not always a single word.

A phrase is defined as a word or a sequence of words that functions as a single unit within a clause/sentence [16, 19]. Given this definition, how would PG handle a phrase that consists of more than one word? For instance,

- (3) Four things scientists have been exploring about the incredibly awesome moon.

In sentence (3), the word group “four things”, and “scientists” are NPs; “have been exploring about the incredibly awesome moon” is a VP which consists of a finite verb group, and a prepositional phrase. From the above description, one can say that the NP, “scientists” being at subject position carries a *sup* in term of subject position. So, “scientists” is treated as a *pada*. The verb group “have been exploring” contains “explore” as a verb and “have.been.ing” as a *tin*. So, “have been exploring” is treated as a *tinanta pada*. But what about the NP “four things” which consists of more than one word and does not have any preposition to mark it a *pada*? And, what about the prepositional phrase “about the incredibly awesome moon” which though has a preposition/*sup* but at the same time contains one more phrase in it, the adjectival phrase (ADJP) “incredibly awesome”? How many *padas* should we consider in the constituents “four things” and “about the incredibly awesome moon”? In order to resolve such issues, let us first look at the characteristics of Sanskrit compounds (*samasta-pada*).

4.1 Four Characteristic Features of the Compounds (*samasta-pada*)

A Sanskrit compound has following four characteristic properties:

1. *Sublopa* (elision of internal *sup/vibhakti*): Elision of internal *sup* as opposed to an *asamasta-pada* (sentence) takes place in a *samasta-pada*. An un-compounded word group is also called *vākya* (sentence) in Sanskrit. Only the final element receives case inflection in compounds. For example, the genitive case marker (*ṣaṣṭhī-vibhakti*) is deleted in the *samasta-pada* *rājapurūṣaḥ* (king-man), whereas in the un-compounded word group *rājñāḥ puruṣaḥ* (king’s man), the genitive case marker is not deleted.
2. *Avyavahāna* (no intervention by other word (*pada*)): Intervention by any other word (*pada*) does not take place in a *samasta-pada* but in an un-compounded word group it can take place. For example, one can say *rājñāḥ ṛddhasya puruṣaḥ* (man of a rich king), where *rājñāḥ puruṣaḥ* has been intervened by *ṛddhasya* a modifier of *rājñāḥ*, but in a compound, *ṛddhasya* cannot modify *rājñāḥ*. One cannot say **rāja-ṛddhasya-puruṣaḥ*.
3. *Niyatapaurvāparya* (fixed word order): The words in a *samāsa* occur in a fixed order. But in a sentence the *padas* can occur freely; *rājñāḥ puruṣaḥ* or *puruṣo rājñāḥ*

4. *Aikasvarya* (accent/stress): All the words in a *samasta-pada* have only one accent/stress. For example, in *rājapuruṣaḥ* the stress is on the final vowel / syllable, but in a sentence, *rājñāḥ puruṣaḥ*, both the words are stressed independently.

Having looked at the characteristic properties of *samasta-pada* (compounds), let us now look at the English phrases and see whether they are comparable with *samasta-pada* or not. Most phrases have all the four properties listed above. We will examine them one by one. For example,

No internal preposition/*sup* is present between the words in the phrase “four things”. Thus, the absence of a *vibhakti* suggests that the internal preposition must have been deleted. So, there is a *sublopa*.

No other phrase can occur within a phrase, i.e. we cannot say “*four to explore things”. So, the phrase has *avyavadhāna* feature of compounds.

The words have fixed order. One cannot say “things four”. So, they follow the principle of *niyatapaurvāparya*.

In English, stress is on the first word in common phrases and on the noun in descriptive phrases. Table 1 shows stress variations in common and descriptive phrases. The words in bold have stress in these phrases⁵.

Table 1. Showing stress variations for common and descriptive phrases.

Common phrase	Descriptive phrase
a sports car	a small car

In our example, in the phrase “four things”, the stress is on ‘**things**’, ‘four **things**’. So, there is *aikasvarya*.

Being able to see the similarities between a phrase and *samāsa/samasta-pada* (compound) only tells us that the components of a phrase have some semantic relation among them. But, even for a compound to participate in a sentence, it has to become a *pada*. It should have some *vibhakti* to express its relation with other participants of the sentence.

It seems that the phrase “four things” neither has any overt *vibhakti* nor any generalized *vibhakti*. Then, how would it pass the test of being a *pada*?

In (3), the verb “explore” has two arguments. One argument is represented by the subject “scientists”. But the second argument represented by “four things” is not at the object position. It has moved leftward to the initial position of the sentence for topicalization [16]. The NP “four things” originated at the object position of “explore” leaves a trace at object position which encodes the relation of the moved element. To put it in other words, the topic position also assigns *vibhakti* to the topicalized constituent. Thus, the NP “four things” has its generalized *vibhakti* in terms of topic position and that is why it is a *pada*.

⁵ See “Learn English with Speak Method, URL: www.speakmethod.com/syllablestresssetphrases.html”

In the PP “about the incredibly awesome moon”, it is not the case that an external element/phrase “incredibly awesome” has intervened between “the” and “moon” hence, there is a violation of one of the compounding characteristics called *avyavadhāna*. Rather, it is an example of embedded compounding. Here, first the words “incredibly” and “awesome” form a compound and then the compound “incredibly awesome” forms another compound with the words “the” and “moon”. After that, the compound “the incredibly awesome moon” takes the preposition “about” as a *sup* which makes it a *pada*.

To summarize, we can say that the constituent “scientists” in sentence (3) occurs at the subject position, hence, as stated by Bharati et al. (1996,1998), it carries a ‘generalized vibhakti’ in terms of subject position. The NP “four things” gets its *sup* inflection from its trace at object position. Also, if the constituent “four things” occurs in a prepositional phrase, both the words take only a single preposition as in *I gave you money for four things*. Therefore, the whole group, “four things” will be treated as a single *subanta pada*. The phrase “about the incredibly awesome moon” can also be taken as a *subanta pada* which carries the preposition “about” as a *sup*.

From the above observations, we can say that complex English phrases come close to compound constructions in Sanskrit, except that in Sanskrit, a compound becomes a single word whereas in English, the phrasal components maintain multiple word status. This hardly matters. Because, compound constructions in English cover a whole range of written styles such as one word as in “milkman” (man who delivers milk), hyphenated as in “milk-fever” (disease caused by lack of the calcium contained in milk) and with white space as in “milk bottle” (bottle for containing milk). Hence unlike Sanskrit, *ekapadibhāva* [20] ‘becoming one word’ of more than one words is not a characteristic property of English compounds. In fact in [14] and [13], Giegerich argues that the distinction between compound and phrase is neither necessary nor possible in English. Therefore, simple English phrases can be treated as *padas*.

So, we can say that English phrases share the properties of a compound to some extent. However, unlike Sanskrit compounds, they maintain more than one word status. Thus, they are not compounds in the strictest sense. Therefore, we call them ‘quasi-compounds’ (*ardhasamāsa*).

We have already seen that in (1), the modifier and modified relation between *vīrāḥ* (brave) and *sainikāḥ* (soldiers) is expressed by attachment of the same case marker to both the words. Similarly, if the head *sainika* takes some other case marker to express its relation, the modifier *vīra* also takes the same case marker such as *vīrān sainikān* (to brave soldiers), *vīrebhyaḥ sainikebhyaḥ* (for brave soldiers), *vīrāṇām sainikānām* (of brave soldiers) etc.,. On the other hand, English attaches a single preposition to all the members of the constituent “brave soldiers”: “by brave soldiers”, “for brave soldiers”, “of brave soldiers”. As a result the positions of the constituent members get fixed and phrases are formed.

As explained in Section 1, a *tiñanta pada* is formed by adding *tiñ* inflections to the verbal bases.

As mentioned in Section 4.1, intervention of any external word/*pada* is not allowed in a *samasta-pada* but the finite verb groups are exceptions to this condition, where an adverbial phrase can intervene in a finite verb group. For example, take the expression in (4):

(4) ‘have been slightly changed’

In (4), the adverb “slightly” is embedded in the *tiñanta pada*, “have been changed”. But then, since a *tiñanta pada* is not a compound, other words can intervene.

It should be noted that except adverbs, no other words can intervene in a *tiñanta pada*. That is why, the expressions like: “*have **to operate** been changed”, “*have been **to operate** changed”, “*have been **to Paul** changed”, etc., become ungrammatical.

Identification of *tiñanta padas* helps in forming a verb group whereby translation of verb and its suffixes can be handled properly.

5 Spectrum of Flexibility in Compounds

If we look at the various types of expressions, there appears to be a continuum from ‘flexible’ expressions to completely ‘fixed’ expressions in compounds. For example, in Sanskrit, *januṣāṅdhah* (blind from birth, born blind), *alpānmuktaḥ* (loosed from a little distance), *vācoyuktiḥ* (appropriate speech) etc., are some of the expressions where there is no deletion (*lopa*) of the internal *sup vibhaktis* and the compound meaning is similar to that of the canonical phrasal paraphrase (*vigraha-vākya*) [15].

Presence of a *vibhakti* expresses the relation between words. In compound expressions, internal *vibhaktis* and number information are not so important [2, 25, 26]. The relations among members of a compound are expressed through positions of the words. That is why *vibhaktis* can vanish. Even in expressions like *stokānmuktaḥ* (loosed from a little distance), *goṣucaraḥ* etc., where the internal *vibhakti* is not deleted, we use the same expression *stokānmuktaḥ*, even if we wish to say *stokābhāṃ muktaḥ* (dual) or *stokebhaḥ muktaḥ* (plural) [26].

In *rājapurusaḥ* etc., even after deletion of the internal *sup*, the potency of the *sup* takes place by *pratyayalakṣaṇa* (A.1.1.62), that is, the operations pertaining to a suffix take place even after deletion of the suffix [27]. Therefore, the component *rājan* is treated as a *pada* for morphological operations, as a result elision of the letter *n* by (A. 8.2.7) can be seen on the surface. Expressions like *brāhmaṇakambalaḥ*, *yūpadāru* etc., also undergo similar operations but no effect of the internal *sup* can be structurally seen on the surface. In such expressions, the compound meaning is similar to their components and one can construct synonymous compounds using synonyms. For instance, *rājabhṛtyaḥ* is the paraphrase of *rājapurusaḥ* and *dvījakambalaḥ* is the paraphrase of *brāhmaṇakambalaḥ*.

On the other hand, there are expressions like *kṛṣṇasarpaḥ* which are completely inflexible. These expressions have all the characteristics of compounds

Table 2. Showing flexibility spectrum in Sanskrit expressions

	<i>alukṣamāsa</i>	<i>samāsa</i>	<i>nityasamāsa</i>
	<i>vācoyuktiḥ</i> (appropriate speech)	<i>rājapurusaḥ</i> (king-man)	<i>kṛṣṇasarpaḥ</i> (cobra)
<i>sup</i> -deletion (<i>sublopa</i>)	x	✓	✓
un-interruption (<i>avyavadhāna</i>)	✓	✓	✓
fixed order (<i>nīyatapaurvāparya</i>)	✓	✓	✓
one accent (<i>aikasvarya</i>)	✓	✓	✓
one word	✓	✓	✓
paraphrasing	✓	✓	x

but irrespective of the meanings of the words involved in compounding the expressions give a more specialized meaning. In our example, the compounding words *kṛṣṇa* (black) and *sarpa* (snake) leave their meanings and give the special meaning “cobra”. Paraphrasing is also not possible in such cases. Such compounds are called *nityasamāsa* (completely fixed expressions). Table 2 gives an overview of spectrum in Sanskrit compounds. It illustrates the four afore mentioned compound properties plus ‘one word’, ‘multiple word’ status and ‘paraphrasing’ with examples where, you will notice that as one moves from left to right the degree of flexibility of the expressions varies.

5.1 Flexibility Spectrum in English Compounds

We claim that all languages including English have flexibility spectrum in compounding. In English, the level of variation is similar to Sanskrit or even higher. We have already seen that English phrases fall under ‘quasi-compound’ class. The ‘quasi-compounds’ show the highest degree of flexibility in English.

Instances of *alukṣamāsa* are also found in English. For example, in “kinsman” (a blood relative, especially a male), marksman” (a person who is skilled in shooting at a mark) etc., the compound members “kins” and “marks” are possessive forms “kin’s” and “mark’s” [10]. The possessive suffix “-’s” does not vanish in compound forms.

The compounds like “blackbird” denoting a bird of a particular species [10] fall under *nityasamāsa* class because such compounds give a very specialized meaning different from the compound members.

Table 3 shows the flexibility spectrum in English compounds.

6 Handling Complex Phrases from The Pāṇinian View Point

After finding evidence for simple phrases, it is time to move towards more complex phrases such as “the symbol of mature wisdom”, where the phrase is composed of two different phrases: an NP “the symbol” and a PP “of mature wisdom”. Or other complex phrases such as “the boy who came from Delhi yesterday”, where the noun phrase is composed of an NP and a relative clause. The

Table 3. Showing flexibility spectrum in English expressions

	<i>ardhasamāsa</i>	<i>aluksamāsa</i>	<i>samāsa</i>	<i>nityasamāsa</i>
	a good boy	kinsman, marksman	lawn tennis, bird-cage, football	blackbird
<i>sup</i> -deletion (<i>sublopa</i>)	✓	x	✓	✓
un-interruption (<i>avyavadhāna</i>)	✓	✓	✓	✓
fixed order (<i>nīyatapaurvāparya</i>)	✓	✓	✓	✓
one accent (<i>aikasvarya</i>)	✓	✓	✓	✓
one word	x	✓	✓	✓
multiple word	✓	x	✓	x
paraphrasing	✓	✓	✓	x

clause in its turn has multiple phrases in it. As shown above, all the member phrases of a complex phrase fall under the *subanta* or *tinānta pada* class.

The PP “of mature wisdom” has a direct semantic connection with the NP “the symbol”.

The minimal phrases forming a complex phrase are seen bound together. That is why the tests like movement and substitutions etc. apply to the entire complex phrase and as a result the whole phrase moves bound together or is substituted as a whole and represents a single meaning which we call *ekārtha*. The fact that the entire complex phrase comes under the scope of a single preposition (*sup*) suggests that the entire complex phrase represents one unit.

Since a complex phrase is composed of two or more phrases/*padas* and all the *padas* maintain their *padaness* i.e. the *subantaness* (*subantatva*), one can not say that a complex phrase is a *pada*. The name we’ve given to such phrases is *subantamukhyaviśeṣyaka-ekārthaka-padasamuccaya*. It means a group of phrases (*padasamuccaya*) which has a single meaning (*ekārthaka*) where a *subanta pada* is the head of that particular complex phrase (*subantamukhyaviśeṣyaka*). The hierarchic organization (internal phrasal structure) within a sentence also supports the *subantamukhyaviśeṣyaka-ekārthaka-padasamuccaya* view.

The concept of *vyapekṣā sāmānyā* in Pāṇini is a general principle to capture meaning interdependence in a sentence. It not only captures the meaning interdependence among nominals but also connects the *subanta padas* with the *tinānta pada*. This generality of *vyapekṣā sāmānyā* might seem to be overcovering the VPs also but the element *subantamukhyaviśeṣyaka* in the term *subantamukhyaviśeṣyaka-ekārthaka-padasamuccaya* restricts it only to the *subanta padas*.

Identification of *padas* helps in the demarcation of syntactic units in a sentence. Once the syntactic units are identified, source language to target language generation becomes easy, especially, when target language is morphologically richer than the source language, such as Hindi and English. Similar to English, Hindi also has ‘quasi-compounds’ phenomenon. For instance, let us take the expressions in (5).

- (5) a. acchā baccā
 good.SG,DIR child.SG,DIR
 ‘good child’
- b. acche bacce ke liye
 good.SG,OBL child.SG,OBL for
 ‘for the good child’

In (5-a), neither of the words has any case marker, therefore, both the words are in direct case [5]. But in (5-b), the head *baccā* (child) has the case marker *ke liye* attached to it. So, the head *baccā* (child) is in oblique case. Since no covert case marker is present after the adjective *acche*, it should be in direct case. But that is not true in this case. But if we consider it to be a ‘quasi-compound’, we can say that *sublopa* (elision of internal case marker/*sup*) has taken place in this expression. And, by the principle of *pratyayalakṣaṇa*, the operations pertaining to *sup* takes place. Hence, the modifier *acche* is in the oblique case.

Identification of syntactic units is also helpful in reordering partially free word order language [8].

7 Conclusions

We have analyzed English phrases based on the Pāṇinian perspective. We have defined nominal inflection *sup*, and finite verb inflections *tiñ* for English and compared English phrases with the notion of *pada* and *samasta-pada* in Sanskrit. We have shown that a single word phrase directly corresponds to the concept of *pada* in Sanskrit and a complex English phrase (a phrase consisting of more than one word) corresponds to compounds. The study shows that the insights from Pāṇinian Grammar can be used to analyze any language from information theoretic point of view. We have also briefly shown its effectiveness in machine translation.

Acknowledgments. We are immensely grateful to Prof. Vineet Chaitanya for sharing his insights and for his continuous guidance during this research. We would also like to thank Prof. Amba P Kulkarni for the discussions in the early stage of the research, to Sirisha Manju, Roja Laxmi and Mahalaxmi for their assistance in implementation of the concepts for English-Hindi machine translation, to Pratibha Rani for her comments on an earlier version of the manuscript, and to Prof. Aditi Mukherjee and Prof. Shatrunjay Rawat for their feedback on the final version of the paper. However, we have not yet fully incorporated their comments into this version of the paper, we wish to incorporate them in future versions of the paper.

References

1. Anantpur, A.P.: Anusaaraka: An approach for MT taking insights from the Indian Grammatical Tradition. Ph.D. thesis, University of Hyderabad (2009)

2. Apte, V.S.: The student's guide to Sanskrit composition. Chowkhamba Sanskrit Series Office (1963)
3. Bharati, A., Bhatia, M., Chaitanya, V., Sangal, R.: Paninian grammar framework applied to English. Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur (1996)
4. Bharati, A., Bhatia, M., Chaitanya, V., Sangal, R.: Paninian grammar framework applied to english. South Asian Language Review 8(1), 1–23 (1998)
5. Bharati, A., Chaitanya, V., Sangal, R.: Natural language processing: a Paninian perspective. Prentice-Hall of India New Delhi (1995)
6. Bharati, A., Kulkarni, A.: Information coding in a language: Some insights from pāṇinian grammar. Dhiimahi, Journal of Chinmaya International Foundation Shodha Sansthan I(1), 77–91 (2010)
7. Bharati, A., Kulkarni, A.: 'subject' in English is *abhihita* (2011)
8. Bharati, A., Sukhada, Jha, P., Paul, S., Sharma, D.M.: Applying sanskrit concepts for reordering in MT. In: Proceedings of the ICON2015 (2015)
9. Bharati, A., Sukhada, Sharma, D.M., Paul, S.: Sanskrit and Computational Linguistics, chap. Anusāraka Dependency Schema from Pāṇinian Perspective. D. K. Publishers (2015)
10. Bloomfield, L.: Language. Motilal Banarasidass Publishers Private Limited, Delhi (1994)
11. Dvivedi, K.: Rachanānuvādakaumudī. Vishwavidyalaya Prakashan (1953)
12. Gangopadhyay, M.: The noun phrase in Bengali: Assignment of role and the kāraka theory. Motilal Banarsidass Publishers. (1990)
13. Giegerich, H.: How robust is the compound-phrase distinction? stress evidence from bi-and tripartite constructions in english. Linguistics 2, 65–86 (2008)
14. Giegerich, H.J.: Attribution in English and the distinction between phrases and compounds (2006)
15. Gillon, B.S.: Exocentric (bahuvrīhi) compounds in Classical Sanskrit. In: Proceedings, First International Symposium on Sanskrit Computational Linguistics. pp. 1–12 (2007)
16. Haegeman, L., Guéron, J.: English grammar: A generative perspective. Blackwell Oxford, England (1999)
17. Joshi, S.: Patanjali's vyakarana-mahabhasya. Samarthahnika (1968)
18. Kapoor, K.: Dimensions of Pāṇini Grammar: The Indian Grammatical System. DK Printworld (2005)
19. Kroeger, P.R.: Analyzing grammar: An introduction. Cambridge University Press (2005)
20. Mahavir: Pāṇini as Grammarian: With Special Reference to Compound Formations. Bharatiya Vidya Prakashan (1978)
21. Mahavir: Samartha Theory of Pāṇini and Sentence Derivation. Munshiram Manoharlal Publishers (1984)
22. Sharma, R.N.: The Aṣṭādhyāyī of Pāṇini: Introduction to the Aṣṭādhyāyī as a Grammatical Device, vol. 1. Munshilal Manoharlal Publishers (1987)
23. Singh, J.D.: Pāṇini, his description of Sanskrit: An analytical study of Aṣṭādhyāyī. Munshiram Manoharlal Publishers (1991)
24. Sobin, N.: Syntactic analysis: the basics. John Wiley & Sons (2010)
25. Speijer, J.S.: Sanskrit Syntax. Motilal Banarsidass (1886)
26. Varma, S.: Vyākaraṇa Kī Dārśanika Bhūmikā. Munshiram Manoharlal, New Delhi (1971)
27. Vasu, S.C.: The Aṣṭādhyāyī of Pāṇini. Motilal Banarsidass Publishers (1996)

A Modified Earley Parser for Huge Natural Language Grammars

Sinan Polat, Merve Selcuk-Simsek, and Ilyas Cicekli

Department of Computer Engineering
Hacettepe University
Ankara, Turkey

spolat@aselsan.com.tr {merveselcuk, ilyas}@cs.hacettepe.edu.tr

Abstract. For almost a half century Earley parser has been used in the parsing of context-free grammars and it is considered as a touch-stone algorithm in the history of parsing algorithms. On the other hand, it is also known for being expensive from its time requirement and memory usage perspectives. For huge context-free grammars, its performance is not good since its time complexity also depends on the number of rules in the grammar. The time complexity of the original Earley parser is $O(R^2N^3)$ where N is the string length, and R is the number of rules. In this paper, we aim to improve time and memory usage performances of Earley parser for grammars with a large number of rules. In our approach, we prefer radix tree representation for rules instead of list representation as in original Earley parser. We have tested our algorithm using different number of rule sets up to 200,000 which are all learned by an example-based machine translation system. According to our evaluation results, our modified parser has a time bound of $O(\log(R)N^3)$, and it has 20% less memory usage regarding the original Earley parser.

Keywords: earley parser, computational linguistics, natural language processing

1 Introduction

Grammar parsers first revealed with programming languages. Until Earley's algorithm, the previous algorithms had been able only to parse unambiguous grammars, or subsets of context-free grammars. Revealing his general context-free grammar parser, Earley made a breakthrough. His algorithm became prior to others because it was able to parse left-recursive grammars, handle empty strings in grammars, and work in both ways as top-down and bottom-up. We analyze the details of his algorithm in Section 2.1. Although Earley parser looks good enough to parse every context-free grammar, an attempt to parse a natural language's grammar has not occurred even after years since his algorithm. Some only tried to parse a partial grammar. While parsing a huge grammar with thousands of rules, Earley's algorithm is ineffective due to following reasons:

- It is not good in memory usage:

- Grouping the similar rules is not a part of the algorithm. The same rules may be stored over and over again unnecessarily resulting to consume the memory.
- It is slow in that it may take even days to parse with a big grammar:
 - The algorithm’s mechanism for handling empty string rules in a grammar is not an ideal solution because it is too costly. Earley’s solution required an extra data structure that needs to be dynamically-updated for this step.
 - Earley used linked-lists as data structures in his work. In linked-lists, time-complexity of getting an item equals to $O(n)$, where n is the number of total item count in the list. For such an expensive algorithm considering time bound, $O(n)$ is really huge. Our plan is to store hundreds of thousands of rules in our data structure. Considering this it would not be plausible to prefer lists here.

The performance evaluation of the Earley’s algorithm is handled in Section 2.2.

As seen from the list above to parse a huge grammar, a modified Earley parser is needed. We put the choice on a regular basis:

- We do not use look-ahead tables in our algorithm to get performance. The details about not using look-ahead is examined in Section 3.1.
- For empty right-hand sides (empty strings) we prefer to use a slightly different version of Aycok and Horspool’s well studied work[1] which is explained deeply in Section 3.2.
- As a data structure we prefer to use a modified radix tree over a list. All rules are carried in the tree, so we believe we build an easily comprehensible, memory efficient, and less expensive in time-complexity system as explained in Section 3.3.
- Two natural language grammars are used for evaluation: Turkish, and English. The rules that we use to parse are all coming from an example-based machine translation system [2, 3]. A variety of rules obtained from different data sets with example-based machine translation are parsed in tests, and we get encouraging results.

Throughout the paper, we assume that the reader is familiar with the basic information of language theory, e.g. context-free grammars (CFG), and context-free languages (CFL). The necessary information can be found in [4, 5]. Paper’s flow is as it follows. In Section 2, we briefly describe the standard Earley parser, and its performance. We give the details of our modified parser including radix tree usage in the rule representation in Section 3. Section 4 presents the evaluation of the modified Earley parser, and in Section 5 we examined the related work before us.

2 Earley Parser

2.1 Earley's Algorithm

Earley's algorithm is a dynamic programming algorithm and a chart parser which resembles Knuth's $LR(k)$ algorithm [6]. Given an input string S_1, S_2, \dots, S_n (n refers to the length of string), the algorithm scans it from left to right. After each symbol S_i is scanned, a state set E_i is generated for it. These sets are mostly called as *Earley sets*, so we named it as E . The state of the recognition process is monitored from state sets. On each set E_i , there are productions to show which rule of the grammar is scanned. Each production has one dot (\bullet) to mark the place that is recognized so far from the right-side of the production. Also, productions include a state reminder j to identify from which state this production is composed. Earley's algorithm has three main phases:

SCANNER. This operation takes place if there is a terminal after dot in a state, e.g. $[T \rightarrow \dots \bullet a \dots, j]$

PREDICTOR. This operation starts when there is a nonterminal after dot in the state, e.g. $[T \rightarrow \dots \bullet P \dots, j]$

COMPLETER. The COMPLETER is applicable if in a state the dot is at the end of the production, e.g. $[T \rightarrow \dots aP \bullet, j]$

2.2 Performance Evaluation of the Standard Earley Parser

Earley's algorithm has a time bound of $O(N^3)$ for ambiguous grammars in worst-case where N is the number of characters in the input string [7]. Another important criterion to determine time complexity, however, is the size of the grammar, i.e. the number of rules in the grammar. Earley states that there is a constant C depending on the size of the grammar, and not depending on n [7]. This information leads to the time complexity proportional to CN^3 .

Although Earley does not discuss the properties of the constant C in his work any further, our experiments in Fig. 1 shows that C results a cost of $O(R^2)$ where R is the number of rules in the grammar. That makes the overall complexity of Earley's algorithm $O(R^2N^3)$. In other words, when we double the size of the grammar, the time required to parse a string is multiplied by a factor of four. Our point here is for very large grammars, e.g. a natural language's grammar, using the standard version of Earley parser is not feasible because parsing operation would take very long time.

3 Modified Earley Parser

We examined Earley's parser from three points of view. First two points cover algorithmic approach, and third one is a structural approach. Our goal is both to accelerate the algorithm, and to use the memory more efficiently.

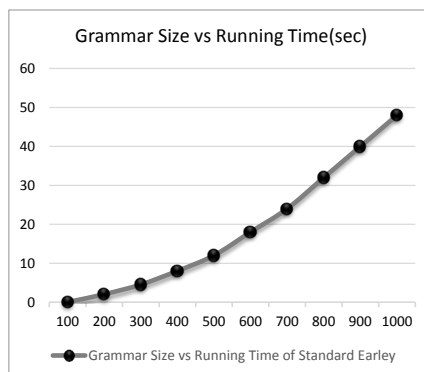


Fig. 1. The running time of the standard Earley parser relating to the number of rules in the grammar

3.1 Look-ahead

Look-ahead principle was one of the features Earley derived from Knuth's work on LR(k) grammars [6, 7]. Earley suggested using look-ahead on COMPLETER step of his algorithm [7]. In time there arose controversy about look-ahead's usage. A research conducted by Bouckaert et al. showed that look-ahead best fit to PREDICTOR part [8]. McLean and Horspool indicated look-ahead slowed down the Earley parser [9]. Later Aycock and Horspool stated look-ahead's necessity was controversial [10]. Because we seek for speed, we decide not to use to look-ahead which defines our parser as an $LR(0)$ parser.

3.2 Empty Sequence (ϵ) Rules

In formal language theory, the empty sequence is a sequence with length zero, and represented by ϵ . If a grammar includes rules with empty strings, parsing the grammar using Earley's algorithm is inefficient. Since Earley parser process items in a particular set of order, encountering an ϵ rule creates problems.

Consider $A \rightarrow \epsilon$ as a rule in the grammar. When A is called in state E_i , it remains without consuming any terminals in E_i . Its staying the same delivers problems, because at this situation COMPLETER will assume $i = k$. However, COMPLETER needs to evaluate the full set of E_k to parse the input correctly. Fortunately, this is a solvable problem. There was some suggestions to fix it including Earley himself [11, 7, 4, 12], yet being either so expensive in time-complexity, or not memory friendly. We prefer to apply Aycock and Horspool's method [1], this simple but effective method explained in Fig. 2.

Although the algorithm in Fig. 2 looks expensive especially considering its first part, it is not a complex operation. Aycock and Horspool didn't present a specific algorithm for this part in their work [1]. Our algorithm for handling the first part (determining all nullable nonterminals in the grammar), doesn't involve

- i Determine all nullable nonterminals in grammar G , e.g. for a nonterminal A :
 $A \Rightarrow^* \epsilon$
- ii Modify the algorithm of the PREDICTOR as follows:
 - for each item $[A \rightarrow \dots \bullet B \dots, j]$ in E_i
 - for each rule $[B \rightarrow \beta]$
 - add $[B \rightarrow \bullet \beta, i]$ to E_i
 - if B is nullable
 - add $A \rightarrow \dots B \bullet \dots, j$ to E_i

Fig. 2. Aycock and Horspool’s algorithm to deal with empty sequences

a loop checking every symbol in the grammar, both terminals, and nonterminals. Because terminals doesn’t have a chance to be empty right-sided, we skip all terminals in a grammar while searching *epsilon* rules. This approach not only decreases the count of the loop, but also reduces the time-complexity of this part. Additionally, the first part is run only for once at the beginning of the flow as a preprocessing step.

3.3 Modified Radix Tree

In our application, we prefer to use radix trees which are slightly different forms of PATRICIA trees [13]. In fact, a radix tree is a special PATRICIA tree with a radix value of 2. Our reasons to use radix trees are:

- *Memory efficiency.* While working with so many rules and huge grammars, efficiently used memory becomes a priority. The space-complexity of a radix tree is $O(CN)$ in that C is the average length of rules, and N is the number of rules in a grammar. One of the best properties of radix trees is its ability to group similar rules. This is an important feature for both memory usage, and compactness of the application.
- *Fast prefix search.* Radix trees’ search time-complexity is $O(N)$.
- *Low insertion complexity.* The insertion complexity of a radix tree is $O(\log(N))$. If it were more expensive than that, we may not be able to construct a structure for hundreds of thousands rules.

The evaluation proof of the radix tree’s performance can be seen in Section 4. We do not cover the deletion of strings from the radix tree here since it is not used by our application. Note that the radix tree structure is generated only once, when the grammar rules are loaded to the parser. After the rules are loaded to the parser, no rule is deleted, so we do not need the deletion operation.

The developed EBMT (Example-Based Machine Translation) system [2, 3] learns translation rules from translation examples between English and Turkish as in Table 1. The learned translation rules can be seen as two natural language grammars. We use modified radix tree to store and benefit our rules learned with our EBMT system. Because we use modified Earley parser in a machine translation environment, radix tree comes in handy both parsing the examples, and matching them for appropriate templates.

Table 1. Learning phase of translation rules

Consider the following translation pairs, in this example there are similar parts in both languages (car and araba, respectively), and also there are differing parts (black-siyah, and red-kirmızı).

black car \Leftrightarrow siyah araba
 red car \Leftrightarrow kırmızı araba

Our EBMT system replaces differing parts with variables to create translation template below:

X^1 car $\Leftrightarrow Y^1$ araba
 if $X^1 \Leftrightarrow Y^1$

Using translation template above, the EBMT system learns the templates below:

black \Leftrightarrow siyah
 red \Leftrightarrow kırmızı

A sample context-free grammar and radix trees between English and Turkish languages are given in Table 2. The specified context-free grammar is composed of learned translation rules. The words between brackets (e.g. $S \rightarrow$ black [*siyah*]) refers to the translation of the source sentence. Since we concern context-free grammars, the presentation involves only a single nonterminal (S) on the left-hand side of all rules. There might be terminals and/or nonterminals on the right-hand sides of the rules. No type information is used in Table 2 to make it more intelligible.

Table 2. Example context-free grammars and radix trees of English and Turkish phrases

English to Turkish	Turkish to English
$S \rightarrow$ black [<i>siyah</i>]	$S \rightarrow$ siyah [black]
$S \rightarrow$ black [<i>kara</i>]	$S \rightarrow$ kara [black]
$S \rightarrow$ red [<i>kırmızı</i>]	$S \rightarrow$ kırmızı [red]
$S \rightarrow$ red [<i>al</i>]	$S \rightarrow$ al [red]
$S \rightarrow$ black hair [<i>siyah saç</i>]	$S \rightarrow$ siyah saç [black hair]
$S \rightarrow$ black hair [<i>kara saç</i>]	$S \rightarrow$ kara saç [black hair]
$S \rightarrow$ black car [<i>siyah araba</i>]	$S \rightarrow$ siyah araba [black car]
$S \rightarrow$ black car [<i>kara araba</i>]	$S \rightarrow$ kara araba [black car]
$S \rightarrow$ black car comes [<i>siyah araba gelir</i>]	$S \rightarrow$ siyah araba gelir [black car comes]
$S \rightarrow$ black car comes [<i>kara araba gelir</i>]	$S \rightarrow$ kara araba gelir [black car comes]


```

graph TD
    S --> black
    S --> red
    black --> hair
    black --> car
    car --> comes
        
```

```

graph TD
    S --> siyah
    S --> kara
    S --> al
    S --> kırmızı
    siyah --> saç
    kara --> araba
    araba --> gelir
        
```

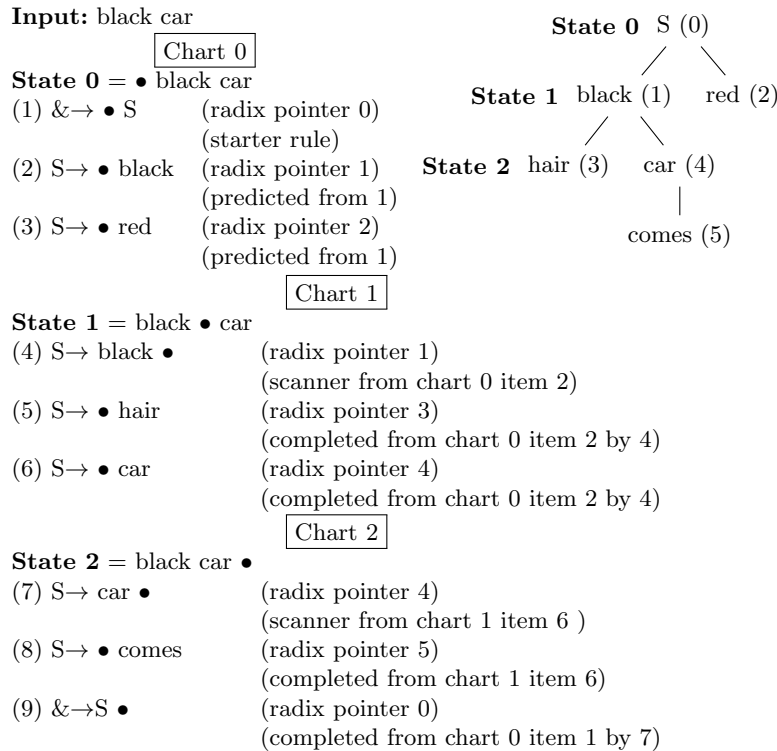


Fig. 3. Modified Earley parser states and radix tree pointers with assigned integer values

The significant points of the modified Earley parser are:

- For each node of the tree we use a vector to carry all the rules belonging to the specific node to be used on translation.
- When building radix trees of grammars, a list of a radix tree pointer is generated for the nodes of tree. Radix tree pointers are used to access a specified node, and its child nodes. Also, integer values are assigned to radix tree pointers to represent these pointers in Earley charts.
- When parsing algorithm gets to PREDICTOR, SCANNER or COMPLETER steps, standard Earley parser adds all rules to the chart according to dot position, yet we don't add rules having the same prefix. Instead of adding all rules to the chart, we add only one rule which contains radix tree pointer relating to the dot position.
- When parsing algorithm gets to PREDICTOR step, radix tree returns the set of rules to make the parsing process faster.
- When parsing algorithm gets to COMPLETER step, the radix tree pointers help to find child nodes to add chart for next dot position.

Similar to the standard Earley parser, we also have parser states and state items in our modified parser. Its internal structure is, however, slightly different

compared to the standard Earley parser. In Fig. 3 modified Earley parser's states, and radix tree pointers with assigned integer values are shown. Integer values of the radix tree pointers are represented in brackets. Parsing the input *black car* is demonstrated according to the given grammar in Table 2.

4 Evaluation

To build various sizes of context-free grammars for evaluation purposes we benefited translation databases that were created by Informatics Association of Turkey (TBD). These databases include about 40,000 of translation pairs. We used different amount of translation pairs from these databases to extract translation templates, and we obtained 200,000 translation templates when all translation pairs are used in the learning phase. The prepared English and Turkish grammars are created from these extracted translation templates. Each grammar includes 200,000 rules and there are approximately 1,080,000 words in each grammar. The average length of the rules for each grammar is 5.4 words and the number of words in a rule is between 1 and 17 words.

The sizes of these grammars vary from 10 rules to 200,000 rules. All the strings used for evaluation are in a language defined by the context-free grammar, i.e. Turkish and English. In order to translate these strings, we parse them by using the standard Earley parser as well as the modified Earley parser. We evaluated average parsing time and memory usage of parsers with randomly selected strings in these datasets. Note that our parser's code is written in Java 1.8, and our evaluation results were taken on Intel i7 processor with 16 GB RAM.

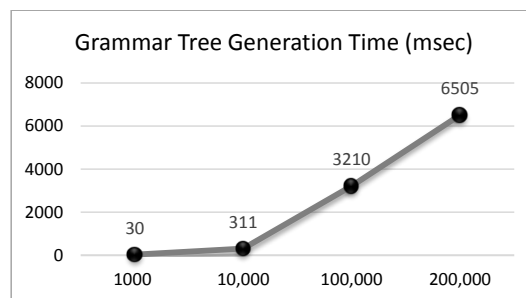


Fig. 4. Complexity of the radix tree generation in relation to the number of rules in the grammar

First, our proposed method requires the generation of the radix tree corresponding to a given context-free grammar. The time-complexity of the radix tree generation for the grammar is $O(N)$ as it is shown in Fig. 4. Generation of the radix tree for a grammar of 10,000 rules takes 311 milliseconds whereas the generation of the radix tree for a 10 times larger grammar of 100,000 rules

takes about 10 times longer, 3210 milliseconds. The generation of the radix tree is required to be performed only once and furthermore even for a grammar with 200,000 rules it takes slightly longer than 6 seconds. It is definitely feasible.

Another important concern is memory efficiency. In order to evaluate memory efficiency of the approaches, we measured total memory usage of the approaches for different size of grammars. According to the results, standard Earley parser is quite conservative in terms of the memory space required for the data structures since all it does is to store the grammar rules in a list. Therefore, in terms of the size of the data structures, the standard Earley parser looks like the most efficient one in our data set as it is shown in Fig. 5. During parsing, however, the standard Earley parser seems to allocate more memory space than our approach, and uses about 64% more memory space than the modified Earley parser. In Fig. 5, standard Earley working set refers to the space requirement of the standard Earley parser for storage of rules, and charts at run time. In the figure the storage requirement of the modified Earley parser is given for both storage of rules and charts. Our approach, the modified Earley parser, uses about twice as much memory as the standard Earley parser to store the rules of the grammar. During parsing our approach uses about 39.25% less memory space than the standard Earley parser. Generally, we think it is fair to say that ours is more efficient in terms of memory usage, compared to the standard Earley parser since we are generally interested in the memory usage patterns during parsing.

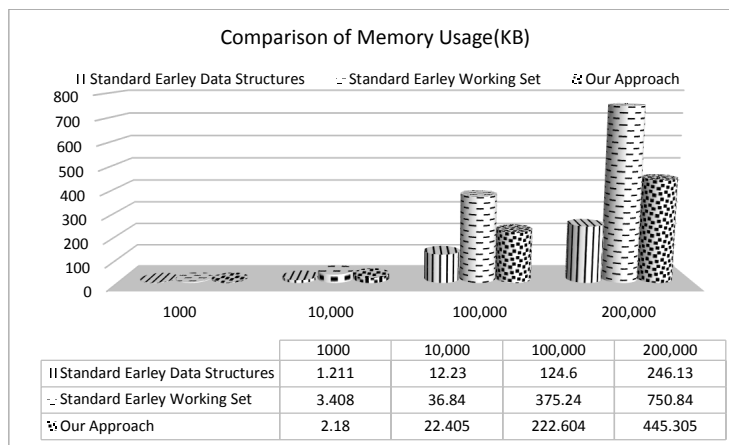


Fig. 5. Comparison of the memory usage of the standard Earley parser’s data structures, standard Earley parser’s working set on running, and our modified Earley parser

The last crucial subject is the running time of parsing. To evaluate running times of the approaches, we measured average running time of the approaches for different size of grammars. According to the evaluated results, the modified Earley parser is significantly faster than the standard Earley parser for very large

grammars. We have shown that the time-complexity of the standard Earley parser is $O(R^2N^3)$ where R is the number of rules in the grammar and N is the string size. The time-complexity of the modified Earley parser, however, is $O(\log(R)N^3)$ where R is the number of rules in the grammar (Fig. 6). Therefore, the modified Earley parser is asymptotically faster than the standard Earley parser. Even if the number of rules in a given grammar are too large, the modified Earley parser can still parse strings quickly. The average time for parsing a string in a grammar with 10,000 rules is about 1.4 seconds, and for a 100,000 rules this increases to 2.2 seconds.

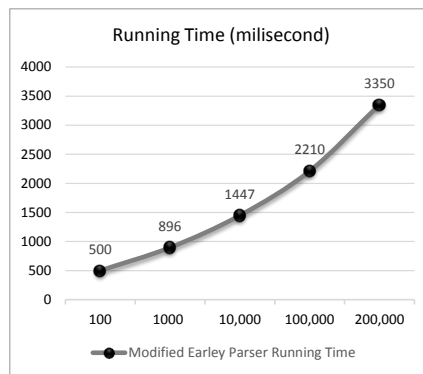


Fig. 6. The average time required to parse a single string in terms of milliseconds, in relation to the number of rules in the grammar

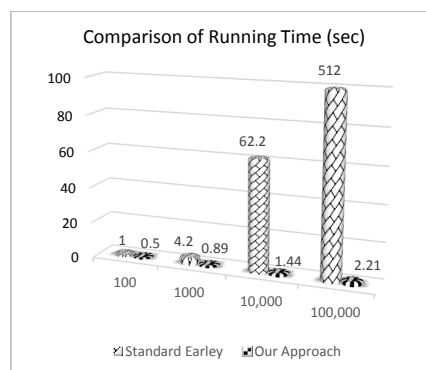


Fig. 7. Comparison between the time-complexities of the standard, and modified Earley parser algorithms in relation to the number of rules in the grammar

The comparison of our parser and the standard Earley parser showed that the modified Earley parser is asymptotically faster than the Standard Earley parser. In fact, as the language gets larger, the gap between the algorithms gets much larger. The comparison for the time-complexities of both parsers is given in Fig. 7. When there are 10,000 rules in the grammar, it takes about 62.2 seconds for the standard Earley parser to parse one string, whereas the modified parser can parse a string in 1.44 seconds. If we then increase the number of rules to 100,000, it takes about 512.0 seconds for the standard Earley parser while the modified parser can parse a string only in 2.21 seconds.

5 Related Work

Graham et al. studied a derivation of Earley's algorithm in their well detailed work [14]. They made a data structure change in the original algorithm of Earley's [7] as we do. They preferred using matrix over list. At the end of their paper, they made a conclusion that their algorithm was less complex than Earley's, and it had worst-case time complexity of $\Omega(n^3)$ where n is the input size.

Tomita [15] examined the base idea of Earley's algorithm, LR parsing algorithm. He generalized the method by precomputing an LR shift-reduce parsing table from a given grammar, and used DAG (Directed Acyclic Graph) as a data structure. He concluded that his algorithm was faster than Earley's algorithm with a five to tenfold speed advantage. Tomita was the first one that testing his system with a natural language grammar. He used a sample English grammar consisting of 220 context-free rules, and 40 sentences.

A faster earley parser was presented by McLean and Horspool [9]. They went for an algorithmic change, used a hybrid approach, and combined Earley's algorithm with LR(k) method (they named it LRE). They made an evaluation using Roskind's *ANSI C* grammar. Their study resulted 10 to 15 times faster recognition, and less than half storage requirements comparing to Earley parser.

Aycock and Horspool conducted a study about zooming in the performance of deterministic and general parsers [10]. They constructed an Earley parser which had speed comparable to deterministic parsers. Radix trees were used in this work to carry parent pointers of items having the same core, but different parents. Trees were constructed if and only if there were two or more Earley items which own the same core with varied parents. In their nodes, trees only carried either 1, or 0. They used *Java* 1.1 grammar with 350 rules to evaluate their systems. Although this study is not directly related to ours, and our tree usage principle is different than theirs as we explained in Section 3.3, we would like to point out that to our knowledge, they were the only ones to use radix trees except us in a study to build faster parsers.

Later Aycock and Horspool proposed a solution to RHS (right-hand side) grammar rules in their inspiring work [1]. They analyzed that Earley's algorithm tackled empty RHS grammar rules, and produced a solution by changing the PREDICTOR step in Earley's algorithm. They tested their parser using programming languages' files, i.e. *python* consisting of 735 files of source code, and

3234 source files from *Java – JDK 1.2.2*. With this solution they made Earley parser 2x speed up.

Horák and Kadlec also used a natural language, Czech language, to evaluate their parsing system, *synt* [16]. In *synt*, a grammar is represented with meta-rule sets for Czech language, while we use learned rules between English and Turkish from our EBMT system. For testing they used 10,000 sentences, and 191,034 words overall; yet we use 200,000 rules for each of our grammars and they contain approximately 1,080,000 words.

6 Conclusion

In this work, we intend to build a faster, and more memory-friendly general context-free grammar parser comparing with the previous parsers, and we introduced our modified Earley parser for this purpose. We profit our parser in an EBMT application and the grammars that we use are induced natural language grammars by our EBMT system, Turkish and English in this context. We run through ϵ rules, and look-ahead tables in Earley’s algorithm. We prefer radix tree as a data structure to carry our learned rules coming from EBMT. 100,000 grammar rules are used to evaluate our work. Tests showed that the bigger a grammar gets, the higher a parser’s time complexity. Our modified Earley parser has a time bound of $O(\log(R)N^3)$ where R is the number of rules in the grammar and N is the string size, and 20% less memory usage. It is asymptotically more efficient than the standard Earley parser in terms of time-complexity, and still slightly more efficient regarding memory usage.

References

1. Aycok, J., Horspool, R.N.: Practical early parsing. *Computer Journal* **45**(6) (2002) 620–630
2. Cicekli, I., Güvenir, H.A.: Learning translation templates from bilingual translation examples. *Applied Intelligence* **15**(1) (2001) 57–76
3. Cicekli, I.: Inducing translation templates with type constraints. *Machine translation* **19**(3-4) (2005) 283–299
4. Aho, A.V., Ullman, J.D.: *The Theory of Parsing, Translation, and Compiling: Volume I: Parsing*. Prentice-Hall, Incorporated (1972)
5. Harrison, M.A.: *Introduction to formal language theory*. Addison-Wesley Longman Publishing Co., Inc. (1978)
6. Knuth, D.E.: On the translation of languages from left to right. *Information and Control* **8**(6) (dec 1965) 607–639
7. Earley, J.: An efficient context-free parsing algorithm. *Communications of the ACM* **13**(2) (1970) 94–102
8. Bouckaert, M., Pirotte, A., Snelling, M.: Efficient parsing algorithms for general context-free parsers. *Information Sciences* **8**(1) (1975) 1–26
9. McLean, P., Horspool, R.N.: A Faster Earley Parser. *Lecture Notes in Computer Science* **1060** (1996) 281–293
10. Aycok, J., Horspool, N.: Directly-executable Earley parsing. *Compiler Construction* (2001) 229–243

11. Earley, J.: An efficient Context-Free Parsing Algorithm. PhD thesis (1968)
12. Jacobs, C., Grune, D.: Parsing techniques: A practical guide. Department of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands (1990)
13. Morrison, D.R.: Patricia—practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM (JACM)* **15**(4) (1968) 514–534
14. Graham, S.L., Harrison, M.a., Ruzzo, W.L.: An Improved Context-Free Recognizer. *ACM Transactions on Programming Languages and Systems* **2**(3) (1980) 415–462
15. Tomita, M.: An efficient augmented-context-free parsing algorithm. *Computational linguistics* **13**(1-2) (1987) 31–46
16. Horák, A., Kadlec, V.: New Meta-grammar Constructs in Czech Language Parser synt. In: *Text, Speech and Dialogue: 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, September 12-15, 2005. Proceedings.* Springer Berlin Heidelberg, Berlin, Heidelberg (2005) 85–92

Extracting HPSG Lexicon from Arabic VerbNet

Noureddine Loukil and Kais Haddar

Faculty of Sciences, University of Sfax, Tunisia
MIR@CL Laboratory, University of Sfax, Tunisia
noureddine.loukil@isimsf.rnu.tn
kais.haddar@fss.rnu.tn

Abstract. This paper presents the construction of a HPSG lexicon of Arabic verbal entities, automatically inferred from the Arabic VerbNet, a large coverage verb lexicon where verbs are classified using syntactic alternations. We discuss the main verb specification along with the relation of the syntactic and semantic levels of representation within the HPSG framework. Extensive analysis of the Arabic VerbNet classes has led to the adoption of a finite set of mapping rules between AVN classes and HPSG subcategorization and semantics descriptions covering the majority of the verbal tokens. We employed the adopted mapping rules to extract the syntactic and semantic data from AVN and finally, we describe the resulting TDL descriptions in which the lexicon has been encoded.

Keywords: Arabic VerbNet, HPSG, Lexicon, Arabic, Automatic Extraction

1 Introduction

The SubCategorization Frame (SCF) of a verb is a specification of the number and type of its complements (objects and oblique arguments) but it evolved to include the specification of the subject in some modern theories like the Head driver Phrase Structure Grammar (HPSG) [1]. In this regard, it becomes more close to the concept of valency introduced in the dependency grammar formalism [2] and generally employed to encompass subcategorization, argument structure, selectional preferences on arguments, and mapping rules which relate syntactic with semantic representation, that is, the syntax semantics interface. SCFs are generally embedded in the syntactic lexicon which is considered among the most valuable resources for many natural language processing tasks [3]. It is widely accepted that the verb encapsulates the main part of a sentence meaning, and thus the description of the syntactic behavior of verbs is a crucial goal for linguists. Thus, creating a syntactic lexicon of verbs with the valency information is obviously substantial for many tasks within the Natural Language Processing (NLP) field.

Several linguistic resources embedding valency information have been created manually and required consequent effort and time. Those resources are mainly

theory-independent so they can be used and shared by several communities such as VerbNet (VN) [4] which is considered to be the largest verb lexicon for English. The Arabic VerbNet [5] is a large coverage verb lexicon where verbs are classified using syntactic alternations [6] and adopting the general structure and content elements of the English VN. Levin's alternations are tested on Arabic verbs but other Arabic-specific alternations are collected. Fine-grained classes of semantic descriptions are identified and verbs are affected to a class based on how they lexicalize the meaning components of the class and how they syntactically realize these components.

Although VN and AVN were designed to be domain and theory independent, many attempts have tried to lexically connect them with other resources such as Propbank [7], WordNet [8], Xtag [9] [10], and FrameNet [11] [12]. In this context, the Semlink project [13] [14] aims to create a mapping between PropBank, FrameNet, WordNet and VN, thus allowing these resources to incorporate and to form an enhanced resource. Moreover, the project includes detailed token mapping of verbs to their VN classes. A set of concept-to-concept mappings are employed to combine the different information provided by these different lexical resources. Although the importance of such project for data enhancement and resource interoperability, it doesn't provide a way to use the SCF and semantics within VN for NLP in theory dependent frameworks. For this reason, we propose to transform the VN valency information to a theory based lexicon so we can effectively use it in tasks such text parsing.

In this paper, we detail the automatic building of a verb lexicon of Modern Standard Arabic (MSA) for the HPSG framework, extracted from AVN verbal classes. The building process was driven by a finite set of mapping rules to transform the linguistic information embedded in AVN to HPSG lexical entries. In the following sections, we present the AVN class anatomy and the HPSG lexical specification and we outline the methodology we developed in which we associate, for each concept within AVN, a parallel concept in the HPSG theory. Mapping rules for syntax and semantics projection are presented and discussed. Finally, we discuss the generated lexicon and we present some issues and some improvements that can enhance the mapping process.

2 Related Work

Arabic is still considered among the less resources languages despite the big interest and the important number of projects that were addressed to it in the recent years. The lack of resources is especially apparent in the field of grammar lexicons. So, manually or automatically developing such resources is of great interest to the Arabic NLP community. [15] reported the development of a manually created SCF resource for Lexical Functional Grammar (LFG) developed for the Arabic LFG parser. It contains 64 syntactic frame, 2,709 lemmas types, and 2,901 lemma frame types, with an average of 1.07 frames per lemma. Additional information about control and specific prepositions with obliques is embedded in the resource. For the HPSG formalism, experimental lexical resource have been

manually developed [16] in TDL within the LKB system. Yet, the coverage of this resource has not been reported.

Automatic acquisition of linguistic resource becomes more than an option by the development of large coverage treebanks for Arabic such as the Penn Arabic TreeBank (PATB)[17]. [18] automatically extract Arabic SCFs by utilizing the automatic Lexical-Functional Grammar (LFG) f-structure annotation algorithm for the PATB developed in [19]. They reported the extraction of 240 frame types for 3,295 lemmas types, with 7,746 lemma frame types (for verbs, nouns and adjectives), averaging 2.35 frames per lemma. There have been also efforts aimed at extracting TAGs from the PATB [20] yet without emphasizing on the lexicon.

3 VerbNet Classes

VerbNet is composed of a list of verb classes. Each VN class contains a set of syntactic descriptions, or syntactic frames (see Fig. 1), containing the possible surface realizations of the argument structure for constructions such as transitive, intransitive, prepositional phrases or resultatives. SElectional Semantic RESTRictions (SELRESTR in AVN) such as “animate” or “human” are used to constrain the types of thematic roles allowed by the arguments, and further restrictions such as syntactic restrictions (SYNRESTR in AVN) may be imposed to indicate the syntactic nature of the constituent likely to be associated with the thematic role. Syntactic frames may also specify the prepositions that are allowed. Each frame is also associated with explicit semantic information, expressed as a conjunction of boolean semantic predicates such as “motion”, “contact”, or “cause”. Each semantic predicate is associated with an event variable E that allows predicates to specify when in the event the predicate is true (start(E) for preparatory stage, during(E) for the culmination stage, and end(E) for the consequent stage).

Table 1 shows a complete entry for a class in Arabic VerbNet 1.1 class \$axara-1 (snort). The 13 verbs in this class use two semantic roles: Agent with selectional restriction “animate” and Theme. The class contains two syntactic frames (intransitive and V NP NP).

Table 1. \$axara-1 class from Arabic VerbNet v1.1

Class: \$axara-1		
Members: 13, Frames: 2		
Members: \$axara, DaraTa, xadafa, EafaTa, xabaja, Eafaqa, EaTasa, fax`a, taja\$`aOa, fasaA, lahava, \$ahaqa, Oaz`a		
Roles:		
Agent [+animate]		
Theme		
Frame	Example	Role assignment
Intransitive	\$axara AlmariyDu. (The patient snorts)	Verb,Agent
V NP NP	\$axara AlmariyDu \$axiyrAF EaAliyAF. (The patient snorts a loud snoring)	Verb,Agent,Theme

```
<FRAME>
<DESCRIPTION primary="V NP fiy deverbal" secondary="deverbal" xtag=""/>
<EXAMPLES>
<EXAMPLE transliteration="$araEa Almuhanodisuwna fiy taxoTiyT AlmaloEab."/>
</EXAMPLES>
<SYNTAX>
  <VERB/>
  <NP value="Agent"> <SYNRESTRS/> </NP>
  <PREP value="fiy"> <SELRESTRS/> </PREP>
  <NP value="Theme">
    <SYNRESTRS>
      <SYNRESTR Value="+" type="deverbal"/>
    </SYNRESTRS>
  </NP>
</SYNTAX>
</FRAME>
```

Fig. 1. AVN Frame from the AVN class badaOa-1, subclass badaOa-1-2

4 The HPSG lexical specification

HPSG [1] is a formal model for the natural language. This model is based on three main components:

- A representation of the linguistic objects as feature structures, making profit of the powerful mathematical apparatus applied on this type of structures and especially the unification operation,
- A specific organization of the space of linguistic objects manipulated by the grammar (the lexicon), designed around the notion of *type*,
- Principles or constraints allowing combination of signs to produce phrases.

The two first component belong to the field of lexicon. In fact, lexical entries are encoded as typed feature structures and types are organized in hierarchies integrated to the lexicon. The HPSG lexicon is designed in extension thus all the forms of a certain lexeme have to be individually specified. Many attempts tried to cope with this redundancy. Flickinger in his dissertation [21] proposed the concept of “lexical rules” to extend the original grammar with diverse rules acting at the lexicon level to automatically infer new lexical entries by inflexion or case altering (for example, deduce passives of verbal entities). This generative tool was very efficient in practice although it was in contradiction with the declarative nature of the targeted grammars.

All the benefits of enforcing feature structures to be typed [22] comes essentially in the verification of grammatical coherence. In fact, in this model, it is impossible to introduce features not appropriated for types. Thereby, all feature structures must be typed. Diverse feature structures are constructed by unification incrementally by applying HPSG Immediate Dominance schemata. Principles (and ID schemata) are considered as the core definition of the grammar. They describe the rules of combination that permit the unification of simple signs to form more complex signs.

4.1 Linguistic objects

Pollard et Sag in [1] adopt a more general vision of the notion of sign proposed by De Saussure [23]. In fact, they consider that the objects of a language (words, phrases or all syntactic constructions) are different types of the same conceptual sign. Thus, each sign is a descriptive construction integrating several linguistic components : phonology, morphology, syntax, semantics and context. The union of all this heterogeneous information makes from the sign a very specific object. Fig. 2 shows the description used for signs of type “ word ”. All the lexemes in a lexicon regardless of their nature obey to this typical specification. They include various simple and complex features more or less used in practice.

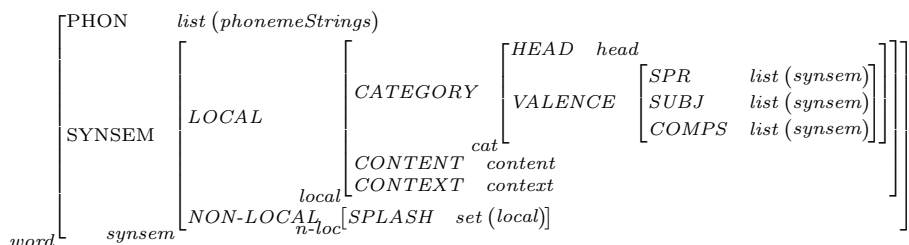


Fig. 2. Typical feature structure of a *word*

- “PHON” : A morphologic description as a value for “ PHONOLOGY ” feature expressed as a list of phonemic tokens.
- “SYNSEM” : Embed the totality of the syntactic and semantic properties of the sign except properties describing unbounded dependencies.
- “LOC” : Describes the sign properties within its actual syntactic construction.
- “NONLOC” : Describes unbounded dependencies that represent relations between the sign and other constituents not belonging to its immediate neighborhood. in this work, We don’t cope with this feature as we express only immediate surface realization of arguments.
- “HEAD” : Describes Part Of Speech properties (type, form, agreement, . . .) along with all other properties without a relation with syntax and semantics.
- “CONTENT” : Describes semantic properties of the sign by specifying the type of the semantic relation and its arguments expressed as a predicative structure.
- “SUBJ” : Describes the subject subcategorized by the sign. Its value is usually composed by just one element.
- “COMPS” : The value of this feature is a list of complements subcategorized by the sign.
- “SPR” : Describes the specifiers subcategorized by the sign.
- “ARG-S” : Describes the argument structure by order of obliquity. It corresponds to the union of feature SUBJ, COMPS and SPR.

- “INDEX” Describes agreement features. Generally used as a reference to the sign.
- “NUCLEUS” describe predicative structure for verbal signs

The VALENCE feature is a specification of the subcategorization of a word. It gives the list of the *synsems* of signs that the word can combine with to make a phrase. the order of those *synsems* is not the surface order but an order of obliqueness defended by many grammatical theories: Subjects first, followed by object complements, prepositional, verbal et adjunct complements.

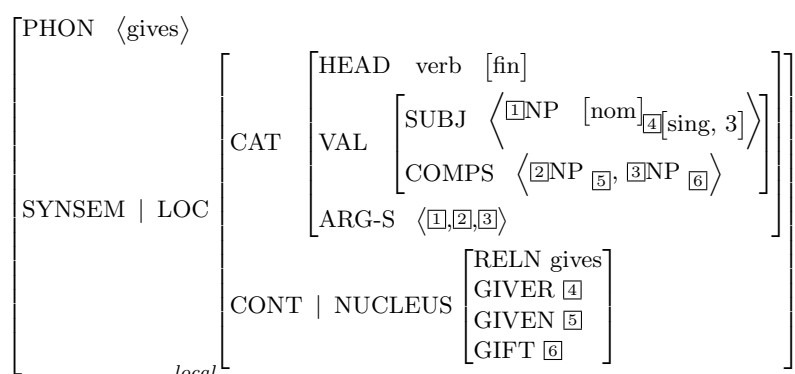


Fig. 3. Lexical Entry of verb “ gives”

The CONTENT Feature in a specification of the semantics of the sign. The representation of semantic relations are borrowed from situational semantics [24]. Semantic roles are assigned within the CONTENT of a lexical entry. A role is assigned by means of structure sharing between the INDEX feature of an element in the SUBJ/COMPS list and the value of some attribute of the sign’s CONTENT value.

4.2 Type Description Language

TDL [25] is a type description language for HPSG. Its syntax provides type definitions organized in hierarchies propagating inheritance relations from root types to instances. Moreover it expresses lexical entry definitions (instances of types) and lexical rules definitions. TDL is a practical language used to implement many HSPG grammars within the LKB [26] system and especially those using the LinGo Grammar Matrix framework [27]. We used a small part of the TDL syntax to define types, subtypes, feature structures, lists and coreferences. A simplified syntax for those definitions is given in Table 2. For reasons of efficiency and interoperability, we declare all the new defined types on top of a subset of the HPSG grammar Matrix types (just the *sign* type as defined in Fig. 4), So the produced resources can be used for processing in the LKB system.

Table 2. TDL definitions

Definition	TDL syntax
Type	$type \rightarrow body\{, option\}^*$
Subtype	$Subtype \rightarrow Supertype \& constraints$
Attribute value Matrix	$avmtype \rightarrow := body\{, option\}^*$
Feature	$feature \rightarrow [attrval\{, attrval\}^*]$
Attribute value	$attribute \rightarrow [attribute\{ : restriction\}]$
List	$list \rightarrow \langle \rangle \langle nonemptylist \rangle$
Coreference	$coreference \rightarrow \#identifier$

$$\begin{aligned}
 synsem &:= avm \& \left[\begin{array}{l} \text{OPT} \quad \text{bool, DEFOPT} \quad \text{bool,} \\ \text{LOCAL} \quad \text{mod-local, NON-LOCAL} \quad \text{non-local-min} \end{array} \right] \\
 sign &:= basic-sign \& \left[\begin{array}{l} \text{SYNSEM} \quad \text{synsem, ARGS} \quad \text{list,} \\ \text{INFLECTED} \quad \text{bool, ROOT} \quad \text{bool} \end{array} \right]
 \end{aligned}$$

Fig. 4. TDL lexical types

5 Mapping Methodology

The method we are using to transform the verb classes provided by AVN to a HPSG lexicon is based on a set of mapping rules in a similar way to the approach followed in the SemLink project [13]. Each rule performs a projection of a particular concept within AVN to a parallel concept within the HPSG lexicon as shown in Table 3. Each AVN verbal class represents several verbs and will be inflated to several HPSG lexical entries. Moreover, the simple HPSG lexical entry represents just one specific syntactic behavior whereas the AVN class may include many syntactic frames. This constrains us to generate multiple different HPSG entries for the same AVN class and even for the same AVN verb. Thereby, we produce $(x \times y)$ HPSG entries starting from an AVN class having x members and y syntactic frames. A mapping rule consists of two parts *IN* and *OUT*. The *IN* specifies a pattern in the AVN class XML. The *OUT* part specifies the corresponding generated TDL type. It can also be a modification in an already generated TDL description. In the conversion phase from AVN to HPSG, the rules are executed. and each time the IN part of a certain rule is identified in an AVN class, a TDL description driven by the OUT part of the rule is generated.

5.1 Class Mapping

The class is the main component within AVN. It encompasses syntax and semantics shared among a set of verbs (the members of the class). The syntax portion of a frame holds argument structure and thematic roles of those arguments which is considered as a semantic information in HPSG. On the other hand, a lexical entry within the HPSG theory is an individual entity that may share properties with other entries through the type inheritance mechanism. The Feature “SYNSEM” of type *synsem* is describing the syntax and semantics of

the lexical entry. Thus, the type *synsem* is the equivalent concept to an AVN class and we set the main mapping rule defined by Rule 1. By the same way, an AVN subclass (used for verbs classes such najaHa-1) is processed in a similar manner as an AVN class. The second mapping rule is defined by Rule 2.

Rule 1 IN $\langle AVNCLASS ID="avn_id" \rangle$
 $\langle FRAME \rangle \langle DESCRIPTION primary="descr" \rangle \dots$
 \Rightarrow **OUT** $avn_id_descr := synsem$

The frame having descr as a value of the description property in an AVN class identified by avn_id is mapped to a subtype of synsem named avn_id_descr:

Rule 2 IN $\langle AVNSUBCLASS ID="avn_sub_id" \rangle$
 $\langle FRAME \rangle \langle DESCRIPTION primary="descr" \rangle \dots$
 \Rightarrow **OUT** $avn_sub_id_descr := synsem$

Table 3. AVN HPSG mappings

AVN Concept	HPSG Concept
Syntactic Frame	synsem Subtype
Verb Member	several HPSG entries
THEME Roles	Thematic roles (CONTENT Feature)
FRAME SYNTAX	Argument structure (ARG-S Feature)
FRAME SEMANTICS	Not used

5.2 Syntax Mapping

An AVN frame syntax as depicted in Fig. 1 describes one of the many possible surface realizations of the argument structure for transitive, intransitive and prepositional phrases among other existing constructions. The feature describing the surface realization of the argument structure within the HPSG theory is the ARG-S. Thus, we map the list of nodes within the SYNTAX, except the $\langle VERB / \rangle$ argument which indicate the verb position, to the ARG-S Feature. The order on the ARG-S list follow exactly the same order of the AVN SYNTAX which also reflect the thematic role order in the AVN THEMEROLES.

The argument list in ARG-S is shared with VALENCE features (SUBJ and COMPS) as shown in the HPSG entry of Fig. 3. We have a degree of freedom in how to share this information because VALENCE does not necessarily show the surface realization. On the other hand, we assume that the first argument, except the verb, represents the subject of the verb and thus is mapped to the SUBJ feature. Indexing is used to relate between the ARG-S argument list and the VALENCE arguments as shown in rule 3.

Rule 3 IN $\langle SYNTAX \rangle \langle ARG1 \rangle \langle /ARG1 \rangle \dots \langle ARGn \rangle \langle /ARGn \rangle$
 $\langle /SYNTAX \rangle \Rightarrow$
OUT $avn_id := synsem \ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ ARG-S \langle \#ind1, \dots, \#indn \rangle]$
 $\ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ VAL \ | \ SUBJ \langle \#ind1 \ ARG1 \ val-arg1 \rangle]$
 $\ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ VAL \ | \ COMPS \langle \#ind2 \ ARG2 \ val-arg2, \dots, \#indn$
 $\ ARGn \ val-argn \rangle]$

Arguments of an AVN class or subclass avn_id are mapped to a subtype of $synsem$ named avn_id whose $SUBJ$ feature takes the first argument and $COMPS$ feature takes the remaining arguments. All arguments are coindexed with $ARG-S$ list.

Syntactic arguments included in the syntactic frame but not playing a thematic role (such as the argument *fiy* in Fig. 1) are not supposed to be considered as complements in the valence HPSG feature (Particularly $COMPS$ and $ARG-S$ feature). Nevertheless, those arguments carry a valuable information about the surface realization of the actual complements. In the mapping process, we choose to map them as specifiers of the arguments following them. Moreover, those arguments are not included on the $CONTENT$ feature. in [1], they are treated as markers. A marker is a functional but not substantial word with *mark* as part of speech. They bear an additional feature called $MARKING$ of sort *marking*. Constituents with marker daughter inherits the $MARKING$ value from that daughter. In our case, as we don't try to specify the constituent itself but a verb, we don't consider the argument preceded by a marker as a *word* but as a phrasal *sign* having a marker daughter. The mapping of a non nominal arguments (such as prepositions) is given in rule 4.

Rule 4 IN $\langle SYNTAX \rangle \dots \langle PREP \ value=valprep \rangle \langle ARGX \rangle \langle /ARGX \rangle \dots$
 $\langle /SYNTAX \rangle \Rightarrow$
OUT $avn_id := synsem \ \& \ [SYNSEM|LOC|CAT \ | \ ARG-S \langle \#ind1, \dots, \#indX \rangle]$
 $\ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ VAL \ | \ COMPS \langle \dots, \#indX \ ARGX \ [MARKING$
 $\ valprep], \dots \rangle]$
The value of prepositional arguments is mapped as a $MARKING$ feature of the following nominal argument.

5.3 Semantics Mapping

The $CONTENT$ feature express the semantics of a lexical entry. For a verb, the $CONTENT$ specifies a semantic relation and the semantic or thematic roles for this relation. In the Fig. 3, the semantic roles $GIVEN$, $GIVER$ and $GIFT$ are fine grained and can be organized in a hierarchy of roles. AVN $THEMEROLES$ does not offer such a fine-grained thematic roles but offer a much generalized thematic roles that can be used within the HPSG lexicon. $THEMEROLES$ are mapped to the $CONTENT$ part of the *sign*. Each $THEMEROLE$ within an AVN class avn_id is mapped to a semantic role having the same name in the $CONTENT$ feature of the HPSG lexical entry as shown in rule 5.

Rule 5 IN `<THEMEROLE> type="th_role" </THEMEROLE> ⇒`
OUT `avn_id := synsem & [SYNSEM | LOC | CONT | NUCLEUS[th_role #reference]]`

Each theme role belonging to theme roles of an AVN class or subclass `avn_id` is mapped to a subtype of `synsem` named `avn_id` whose `NUCLEUS|th_role` feature share its value via a reference with the `CONT|INDEX` feature of the corresponding surface argument in the `COMPS` list. The semantic relation feature `CONT|RELN` takes the name of the concerned verb.

5.4 Restrictions Mapping

Syntactic restrictions are syntactic constraints expressed specially for some argument. They are descriptions of the surface argument filling in the role. As shown in Fig. 5, the restriction is about marking a nominal argument by a certain marker (`Oano_comp` in this example). We simply map the type of the marker as a `MARKING` feature of the argument. This mapping is given by rule 6.

Rule 6 IN `<SYNTAX>... <ARGX><SYNRESTR Value="+" type=restrvalue /></ARGX>... </SYNTAX> ⇒`

OUT `avn_id := synsem & [SYNSEM | LOC | CAT | VAL | COMPS<,...,#indX ARGX [MARKING restrvalue],...>]`

The value of nominal arguments having a syntactic restriction of type x is mapped as the value of the `MARKING` feature of the same argument.

```
<NP value="Theme">
<SYNRESTRS>
<SYNRESTR Value="+" type="Oano_comp"/>
</SYNRESTRS>
</NP>
```

Fig. 5. Syntactic restriction on marker

Syntactic restrictions of type “plural” or “dual” (See Fig. 6) are mapped to the `PERSON` feature included in the `INDEX` feature of the argument specified in the verb’s `COMPS` valency list. Within a frame, syntactic restrictions may inform about the particular surface realization such as “sentential”, “genetive” and “deverbal” which are fine grained sorts of arguments. We simply ignore those types of restrictions for the argument at this level.

6 Implementation and Discussion

We used the mapping rules described in the previous section in order to develop a prototype program called `AVN2HPSG` for the conversion process. `AVN2HPSG` is

```

<NP value="Theme">
  <SYNRESTRS logic="or">
    <SYNRESTR Value="+" type="plural"/>
    <SYNRESTR Value="+" type="dual"/>
  </SYNRESTRS>
</NP>

```

Fig. 6. Syntactic restriction on number

written in C# to benefit from the *LINQ* (Language INtegrated Query)chnology through the use of Linq2XML, a powerful query framework for XML documents that simplify the task of information extraction from AVN based on specified criteria. The mapping rules are applied at specific conditions while the program navigates the XML of each AVN file.

Each XML File in AVN is a definition of an AVN class. AVN2HPSG parse the AVN class and creates two files: a TDL types file consisting of types equivalent to AVN class frames and a second TDL file consisting on verb instance definition and equivalent to AVN verb mermbers. Starting from 783 syntactic frames and 5835 verb members of the Arabic VerbNet 1.1, the program execution produced TDL files containing 555 TDL types and 14323 TDL verb specifications. There was a loss of information due to frames not conforming to the rules input specifications or simply because we ignored some cases

In order to keep the resulted resource as coherent as possible, we made some conditions to eliminate some information that may generate confusing TDL descriptions. For example, AVN contains frames in which verbs are not the first argument. Whereas, they are representing practical SCF, those frames describing nominal sentences in Arabic languages shouldn't be mapped like they are presented but firstly converted to equivalent verbal SCF.

We note a certain redundancy in the produced TDL specifications because of the absence of a type hierarchy. In fact AVN is not designed to capture syntactic generalization but semantic ones. So classes represent distinctive semantics but tend to always repeat the same argument structure descriptions. We plan to cope with this redundancy by enhancing the proposed method to automatically create a special type hierarchy to eliminate this issue.

The proposed methodology and the developed prototype program was applied on AVN but there is no assumptions that was made prohibiting the evaluation for the English VN. We plan to map the English VN to HPSG so wa can have more scope on evaluating the output lexicon compared to well elaborated HPSG lexica for English.

Though the generated lexicon is in TDL and can be directly used in LKB system, we plan to adopt the TDL type hierarchy employed in matrix grammars which can yield to more organized lexicon and easy operation by grammar writers.

7 Conclusion

We detailed the process of building a HPSG lexicon from Arabic VerbNet. We have successfully extracted a verbal lexical resource for the HPSG formalism expressed in TDL to offer an easy way to use it in within the LKB system. The entire process is driven by specific mapping rules that can be enhanced to extract more information into the target lexicon. We think this resource is of great value for natural language processing and for evaluating purposes against other automatically and manually constructed resources for Arabic. We plan to include more organization in the generated lexicon by using an adapted TDL type hierarchy. Moreover, since AVN integrate deverbal and participle lemma for each verb, we plan to extract those lexemes and associate them syntactically and semantically with their corresponding verbs to build a large coverage lexicon.

References

1. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. University of Chicago Press, Chicago (1994)
2. Tesnière, L.: *Eléments de syntaxe structurale*. Klincksieck, Paris (1959)
3. Briscoe, T., Carroll, J.A.: Automatic extraction of subcategorization from corpora. In: ANLP. (1997) 356–363
4. Schuler, K.K.: VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. PhD thesis, University of Pennsylvania (2005)
5. Jaouad, M.: Classifying arabic verbs using sibling classes. In: In the Proceedings of the International Workshop on Computational Semantics, Oxford, UK (2011)
6. Levin, B.: *English Verb Classes and Alternations: a preliminary investigation*. University of Chicago Press, Chicago and London (1993)
7. Palmer, M., Kingsbury, P., Gildea, D.: The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics* **31** (2005)
8. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Wordnet: An on-line lexical database. *International Journal of Lexicography* **3** (1990) 235–244
9. Doran, C., Egedi, D., Hockey, B.A., Srinivas, B., Zaidel, M.: Xtag system – a wide coverage grammar for english (1994)
10. Ryant, N., Kipper, K.: Assigning xtag trees to verbnet. In: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms, Vancouver, Canada (2004)
11. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: In proceedings of the COLING-ACL. (1998) 86–90
12. Pazienza, M.T., Pennacchiotti, M., Zanzotto, F.M.: Mixing wordnet, verbnet and propbank for studying verb relations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC-2006). (2006)
13. Palmer, M., Bonial, C., Mccarthy, D.: Semlink+: Framenet, verbnet and event ontologies. In: ACL2014, Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014). (2014)
14. Loper, E., ting Yi, S., Palmer, M.: Combining lexical resources: Mapping between propbank and verbnet. In: In Proceedings of the 7th International Workshop on Computational Linguistics. (2007)

15. Attia, M.: Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation. PhD thesis, The University of Manchester, Manchester, UK (2008)
16. Boukedi, S., Haddar, K.: HPSG grammar for arabic coordination experimented with LKB system. In: Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, May 21-23, 2014. (2014)
17. Maamouri, M., Bies, A.: Developing an arabic treebank: Methods, guidelines, procedures, and tools. In: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. Semitic '04 (2004) 2–9
18. Attia, M., Shaalan, K.F., Tounsi, L., van Genabith, J.: Automatic extraction and evaluation of arabic LFG resources. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012. (2012) 1947–1954
19. Tounsi, L., Attia, M., Genabith, J.V.: Automatic treebank-based acquisition of arabic lfg dependency structures. In: Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages. Semitic '09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 45–52
20. Habash, N., Rambow, O.: Extracting a tree adjoining grammar from the penn arabic treebank. In: In Traitement Automatique du Langage Naturel (TALN-04. (2004)
21. Flickinger, D.: Lexical Rules in the Hierarchical Lexicon. PhD thesis, Linguistics Department, Stanford University, Stanford, California (1987)
22. Blache, P.: Constraints, linguistic theories and natural language processing. In: Natural Language Processing - NLP 2000, Second International Conference, Patras, Greece, June 2-4, 2000, Proceedings. (2000) 221–232
23. Saussure, F.D.: Cours de linguistique générale. v.C. Bally and A. Sechehaye (eds.), Paris/Lausanne (1916) English translation: Course in General Linguistics. London: Peter Owen, 1960.
24. Barwise, J., Perry, J.: Situations and Attitudes. The MIT Press: Cambridge, MA, USA (1983)
25. Krieger, H.U.: TDL - A Type Description Language for HPSG. Part 2: User Guide. DFKI, Saarbruecken. (1994) Research Report D-94-14.
26. Copestake, A.: Implementing Typed Feature Structure Grammars. CSLI Publications, Stanford, CA (2002)
27. Bender, E.M., Drellishak, S., Fokkens, A., Goodman, M.W., Mills, D.P., Poulson, L., Saleem, S.: Grammar prototyping and testing with the lingo grammar matrix customization system. In: ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, System Demonstrations. (2010) 1–6

A Survey of Arabic Text Representation and Classification Methods

Rami Ayadi¹, Mohsen Maraoui², Mounir Zrigui³

¹ LaTICE laboratory, University of Sfax, Sfax, Tunisia

ayadi.rami@planey.tn

² Computational Mathematics Laboratory, University of Monastir, Monastir, Tunisia

maraoui.mohsen@gmail.com

³ LaTICE laboratory, Faculty of science of Monastir, Monastir, Tunisia

mounir.zrigui@fsm.rnu.tn

Abstract. In this paper we have presented a brief current state of the Art for Arabic text representation and classification methods. First we describe some algorithms applied to classification on Arabic text. Secondly, we cite all major works when comparing classification algorithms applied on Arabic text, after this, we mention some authors who proposing new classification methods and finally we investigate the impact of preprocessing on Arabic TC.

Keywords: Arabic, impact of preprocessing, Text classification.

1 Introduction

Many researchers have been worked on text classification in English and other European languages such as French, German, Spanish, and in Asian languages such as Chinese and Japanese. However, researches on text classification for Arabic language are fairly limited.

The text classification problem is composed of several sub problems, which have been studied intensively in the literature such as the document indexing, the weighting assignment, document clustering, dimensionality reduction, threshold determination and the type of classifiers. Several methods have been used for text classification such as: Support Vector Machines (SVMs), K Nearest Neighbor (KNN), Neural Networks (NN), Naïve Bayes (NB), Decision Trees (DT), Maximum Entropy (ME), N-Grams, and Association Rules.

Term indexing and weighting aim to represent high quality text. High quality in text mining usually refers to some combination of relevance, novelty, and interestingness.

Several approaches have been used to index and weight terms but all of them share the following characteristics [1]: The more the number of times a term occurs in documents that belong to some category, the more it is relative to that category.

The more the term appears in different documents representing different categories, the less the term is useful for discriminating between documents as belonging to

different categories. The most commonly used weighting approach is the Term Frequency Inverse Document Frequency tf-idf.

2 Related work

Researches on the field of Arabic TC fall into two categories: applying and comparing classification algorithms on Arabic text, and investigates the impact of dimensionality reduction.

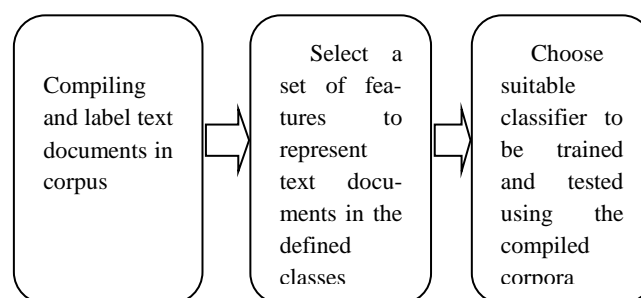


Fig. 1. Building Text Classification System Process

2.1 Classification and Comparing Algorithms on Arabic Text

El Koudri [2] classified Arabic web documents automatically by Naive Bayes (NB) which is a statistical machine learning algorithm, is used to classify non-vocalized Arabic web documents (after their words have been transformed to the corresponding roots) to one of five pre-defined categories. Cross validation experiments was used to evaluate the NB categorizer. Elkoudri used a corpus of 1500 text documents belonging to 5 categories each category contains 300 text documents. With 2,000 terms/roots, the categorization accuracy varies from one category to another with an average accuracy over all categories of 68.78 %. Furthermore, the best categorization performance by category during cross validation experiments goes up to 92.8%.

Maximum entropy (ME) used by El-Halees [3] and Sawaf [4] for classifying Arabic text documents to classify news articles. The first author preprocessed data using natural language processing techniques such as tokenizing, stemming and part of speech then he used maximum entropy method to classify Arabic documents. The best classification accuracy reported was 80.41% and 62.7% by Sawaf using statistical methods without any morphological analysis.

Al Zoghby [5] introduce a new system developed to discover soft-matching association rules using a similarity measurements based on the derivation feature of the Arabic language. In addition, He presents the features of using Frequent Closed Itemsets (FCI) concept in mining the association rules rather than Frequent Itemsets (FI).

Meshleh [6] implements a Support Vector Machines (SVMs) based text classification system for Arabic language articles using CHI square method as a feature selection method in the pre-processing step of the Text Classification system design proce-

ture. Comparing to other classification methods, the system shows a high classification effectiveness for Arabic data set in term of F-measure (F=88.11). He used an in-house collected corpus from online Arabic newspaper archives, including Al-Jazeera, Al-Nahar, Al-Hayat, Al-Ahram, and Al-Dostor as well as a few other specialized websites. The collected corpus contains 1445 documents that vary in length. These documents fall into nine classification categories. Also in another work [7], he investigates the effectiveness and performance of six (CHI, NGL, GSS, IG, OR and MI) commonly used feature selection methods with SVMs evaluated on an Arabic dataset, he concludes that CHI, NGL and GSS performed most effective with SVMs for Arabic TC tasks.

Harrag in [8] presents the results of classifying Arabic text documents using a decision tree algorithm. The results show that the suggested hybrid approach of Document Frequency Thresholding using an embedded information gain criterion of the decision tree algorithm is the preferable feature selection criterion. The study concluded that the effectiveness of the improved classifier is very good and gives generalization accuracy about 0.93 for the scientific corpus and 0.91 for the literary corpus. Experiments are performed over two self-collected data corpus; the first one is from the scientific encyclopedia “Do You Know” (هل تعلم). It contains 373 documents belonging to 1 of 8 categories (innovations, geography, sport, famous men, religious, history, human body, and cosmology), each category has 35 documents. The second corpus is collected from Hadith encyclopedia (موسوعة الحدّث الشرف) from “the nine books” (الكتب التسعة). It contains 435 documents belonging to 14 categories.

The key Nearest Neighbor (kNN) algorithm, which is known to be one of top performing classifiers applied for the English text along with the Support Vector Machines (SVMs) algorithm, has been implemented by Al-Shalabi [9] to the problem of Arabic text categorization. He used Document Frequency threshold (DF) method to keyword extraction and reduction dimensionality. The results show that kNN is applicable to Arabic text; has been reached a 0.95 micro-average precision and recall scores, using a corpus from newspapers (Al-Jazeera, An-Nahar, Al-Hayat, Al-Ahram, and Ad-Dostor) and from Arabic Agriculture Organization website. The corpus consists of 621 documents belonging to 1 of 6 categories (politics 111, economic 179, sport 96, health and medicine 114, health and cancer 27, agriculture 100). They pre-processed the corpus by applying stopwords removal and light stemming.

Kheirsat in [10] presented a machine learning approach for classifying Arabic text documents. For the problem of high dimensionality of text documents, embeddings are used to map each document (instance) into R (the set of real numbers) representing the tri-gram frequency statistics profiles for a document. Kheirsat classifies a test text document by computing Manhattan/Dice distance similarity measure to all training documents and assign the class of the training document with smallest/largest computed distance to the test text document. Kheirsat concluded that classification using the Dice measure outperformed classification using the Manhattan measure. Although the Manhattan measure has provided good classification results for English text documents, it does not seem to be suitable for Arabic text documents. Kheirsat collected her corpus from Jordanian Arabic newspapers (Al-Arab, Al-Ghad, Al-Ra’I, Ad-Dostor). The corpus consisted of text documents covering four categories: sports,

economy, technology and weather. The technology and weather documents were very small in size ranging from 1 to 4 KB. Sports and economy documents were much larger ranging from 2 to 15KB for sports documents and 2 to 18KB for economy documents. The smaller documents constituted about 2% of the total number of documents in the sports and economy category. She applied stop words removal and used 40% for training and 60% for testing.

Harrag in [11] proposes the application of Artificial Neural Network for the classification of Arabic language documents. An Arabic corpus is used to construct and test the ANN model and he discussed the methods of document representation, assigning weights that reflect the importance of each term. Each Arabic document is represented by the term weighting scheme. As the number of unique words in the collection set is big, the Singular Value Decomposition (SVD) has been used to select the most relevant features for the classification. The experimental results show that ANN model using SVD achieves 88.33% which is better than the performance of basic ANN which yields 85.75% on Arabic document classification.

Some studies are compared classification algorithms on Arabic text. Hmeidi reported in [12] a comparative study of two machine learning methods on Arabic text categorization. He evaluated K nearest neighbor (KNN) algorithm, and support vector machines (SVM) algorithm using the full word features and considered the tf.idf as the weighting method for feature selection, and CHI statistics as a ranking metric. Experiments showed that both methods were of superior performance on the test corpus while SVM showed a better micro average F1 and prediction time. The used training and testing data sets are subsets of the most common newspapers in Jordan which are called Alrai and Addustour newspapers. The number of training articles was 2206 articles, and the number of testing articles is 29 articles. The collected documents belong to one of two categories (sport and economic).

In [13] Abbas compared Triggers Classifier (TR-Classifier) and KNN to identify Arabic topic. Performances are acceptable, particularly for TR-classifier, using reduced sizes of vocabularies. For the TR-Classifier, each topic is represented by a vocabulary which has been built using the corresponding training corpus. Whereas, the kNN method uses a general vocabulary, obtained by the concatenation of those used by the TR-Classifier.), the average recall and precision for KNN and TR are 0.75, 0.70 and 0.89, 0.86 respectively. Abbas collected 9,000 articles from Omani newspaper (Al-Watan) of year 2004. The corpus belongs to 1 of 6 categories (culture, economic, religious, local news, international news). The corpus includes 10M word including stopwords. After removing stopwords and infrequent words the vocabulary size became 7M words. Tf.idf was used as weighting schemes.

Duwairi in [14] compared the performance of three classifiers for Arabic text classification as KNN, Naïve Bayes, and Distance-Based classifier. Each documents were preprocessed by removing punctuation marks and stop words, then all of them are represented as a vector of words (for the case of Naïve bayes, he used a vector of words and their frequencies). As a method to reduce the dimensionality of feature vector, the author use Al-shalabi stemmer. Experimental results show that NB outperforms the other two algorithms. Duwairi collected 1,000 documents fall into 10 predefined categories; each category contains 100 documents. The set of categories in-

clude: sport, economic, internet, art, animals, technology, religious, politics, medicine and plants.

In another work, three classification algorithms on Arabic text are compared by Kannan in [15]; the three algorithms were KNN, NB, and Rocchio. The research results reveal that Naïve Bayes was the best performer ($F1=0.8209$), followed by kNN ($F1=0.7871$) and Rocchio ($F1=0.7882$). The used corpus is collected from online newspapers (Al-Jazeera, An-Nahar, Al-Hayat, Al-Ahram, and Ad-Dostor). The corpus consists of 1445 documents belonging to 9 categories (medicine, sport, religious, economic, politics, engineering, law, computer, and education). They applied light stemming approach for feature reduction and 4-folds cross-validation was performed for evaluation.

The performance of two popular text classification algorithms (SVMs and C5.0) is evaluated in [16] by Al-Harbi to classify Arabic text using seven Arabic corpora. The average accuracy achieved by SVMs is 68.65%, while the average accuracy achieved by C5.0 is 78.42%. One of the goals of their paper is to compile Arabic corpora to be benchmark corpora. The authors compiled 7 corpora consisting of 17,658 documents and 11,500,000 words including stopwords but the corpora are not available publically.

Bawaneh applied KNN and NB on Arabic text and conclude that KNN has better performance than NB [17], they also conclude that feature selection and the size of training set and the value of K affect the performance of classification. The Researchers also posed the problem of unavailability of freely accessible Arabic corpus. The in-house collected corpus consists of 242 documents belonging to 6 categories. Authors applied light stemming as a feature reduction technique and tf-idf as weighting scheme, they also performed cross-validation test.

Alsalem in [28] investigate Naïve Bayesian method (NB) and Support Vector Machine algorithm (SVM) on different Arabic data sets for TC. The data used are The Saudi Newspapers (SNP) [1], the data set consist of 5121 Arabic documents of different lengths that belongs to 7 categories, the categories are (Culture "الثقافية", Economics "الإقتصادية", General "العامة", Information Technology "تكنولوجيا المعلومات", Politics "السياسية", Social "الأجتماعية", Sport "الرياضة"). The average of three measures obtained against SNP Arabic data sets indicated that the SVM algorithm ($F1=0.778$) outperformed NB algorithm regards to $F1=0.74$, Recall and Precision measures.

El-Halees in [18] compared six well known classifiers, which are: Maximum entropy, Naïve Bayes, Decision Tree, Artificial Neural Networks, Support Vector Machine, and k-Nearest Neighbor using the same data sets and the same experimental settings. The recall, precision and f-measure for the classifiers are computed and compared. The author compared the methods after preprocessing and all stop words are removed and he found that the performance of Naïve Bayes is the best ($F1=91.81$), the performance of Maximum Entropy, Support Vector Machine and Decision Tree are acceptable, but the performance of k-Nearest Neighbor and Artificial Neural Networks was bad.

However, after using Information Gain as feature selection, the data was reduced significantly and the performance of k-Nearest Neighbor and Artificial Neural Networks improved significantly. The performance of Naïve Bayes did not change but

still the best classifier to Arabic corpus.

In these experiments, the author used an Arabic documents collected from Aljazeera Arabic news channel. The documents categorized into six domains: politics, sports, culture and arts, science and technology, economy and health. The author applied stop words removal and normalization and used 10-folds cross-validation for testing.

Ismail in [29] implemented the Sequential Minimal Optimization (SMO), Naïve Bayesian (NB) and J48 (C4.5) Algorithms using weka¹ program and compared between the algorithms in accuracy and time to get the result. The data set used consists of 2356 documents of different lengths. Each document was manually labeled based on its contents and the domain that it was found within, these documents categorized to six categories (Sport, Economic, Medicine, Politic, Religion and Science) where 60% of the data used as training and the remaining 40% used as testing. Token reduction approach for documents is used to minimize storage requirements and all the types of stop words are removed. The results show that the Sequential Minimal Optimization (SMO) classifier achieves the highest accuracy (96.08%) and the lowest error rate (3.42%), followed by the J48 (C4.5) classifier (90.48% and 9.52%), then by the Naive Bayes (NB) classifier (85.60% and 14.4%). The second part of the results shows that the time needed to build the SMO model is the faster one (5.2 seconds), followed by NB model (12.0 seconds), then J48 classifier which takes a highest amount of time (291.72 seconds).

Table 1. F_mesure

	F-measure without reduction	F-measure with Information Gain
Maximum entropy	85.96	83.83
Naïve Bayes	91.81	83.9
Decision Tree	71.91	74.48
Artificial Neural Networks	10.81	74.33
Support Vector Machine	88.33	88.33
k-Nearest Neighbor	38.6	70.07

Duwairi in [19]- [20] propose a distance-based classifier for categorizing Arabic text. Each category is represented as a vector of words in an m-dimensional space, and documents are classified on the basis of their closeness to feature vectors of categories. The classifier, in its learning phase, scans the set of training documents to extract features of categories that capture inherent category specific properties; in its testing phase the classifier uses previously determined category-specific features to categorize unclassified documents. Stemming was used to reduce the dimensionality of feature vectors of documents. The accuracy of the classifier was tested by carrying out several categorization tasks on an in-house collected Arabic corpus. The average accuracy reported was 0.62 for the recall and 0.74 for the precision. He collected 1000

¹ <http://www.cs.waikato.ac.nz/ml/weka/>

text documents belonging to 10 categories (sport, economic, internet, art, animals, technology, plants, religious, politics, and medicine). Each category contains 100 documents. She used 50% for training and 50% for testing.

Alruily describes an initial prototype for identifying types of crime in a Arabic text within the crime domain. Two approaches are explored to perform recognition tasks. The first approach completely relies on direct recognition using gazetteers. The second approach is a rule-based system. Rules are built based on the predefined crime indicator list that contains some important keywords [21].

Abbas in [22] proposed Triggered (TR) classifier. Triggers of a word W_k are ensemble of words which highly correlated with it. The main idea of TR-Classifier is computing the average mutual information (AMI) for each couple of words from the training documents and testing document, and then assigns the topic that highest AMI to the test document. The best recall achieved is 0.9.

In [23], Ayadi applied inter-textual distance theory to classify any anonymous Arabic text according to criteria of lexical statistic; this requires integration of a metric for classification task using a database of lemmatized corpus.

Syiam presented an Arabic text categorization system based on Machine learning algorithms and many algorithms for stemming and feature selection [24]. The document is represented using several term weighting schemes and finally the k-nearest neighbor and Rocchio classifiers are used for classification process. Experiments show that the hybrid method of statistical and light stemmers is the most suitable stemming algorithm for Arabic language and the hybrid approach of document frequency and information gain is the preferable feature selection criterion and normalized-tfidf is the best weighting scheme. Finally, Rocchio classifier has the advantage over k-nearest neighbor classifier in the classification process and gives generalization accuracy of about 98%.

2.2 The Impact of Dimensionality Reduction in TC

Duwairi in [25] analyzed and compared three feature reduction techniques; stemming, light stemming, and word clusters using K-nearest-neighbor classifier applied to Arabic text. The purpose of employing the previous methods is to reduce the size of document vectors without affecting the accuracy of the classifiers. Comparison metrics are size of document vectors, classification time, and accuracy (in terms of precision and recall). The corpus consists of 15,000 documents belonging 3 categories: sports, economics, and politics. In terms of vector sizes and classification time, the stemmed vectors consumed the smallest size and the least time necessary to classify a testing dataset that consists of 6,000 documents. The light stemmed vectors superseded the other three representations in terms of classification accuracy.

Thabtah tested and compared three variations of vector space models (VSMs) (these variations are Cosine coefficient, Dice coefficient and Jaccard coefficient) and term weighting approaches (IDF, WIDF, ITF and $\log(1+tf)$) using KNN algorithm [26]. The Experimental results on different Arabic text categorization data sets collected from online Arabic newspapers including Al-Jazeera, Al-Nahar, Al-hayat, Al-Ahram, and Al-Dostor. With regards to F1 results, the author concluded that Dice

based TF.IDF (94.91) and Jaccard based TF.IDF (94.91) outperformed Cosine based TF.IDF (90.93), Cosine based WIDF (75.94), Cosine based ITF (91.02), Cosine based $\log(1+tf)$ (92.65), Dice based WIDF (81.01), Dice based ITF (89.63), Dice based $\log(1+tf)$ (85.21), Jaccard based WIDF (81.01), Jaccard based ITF (89.63), and Jaccard based $\log(1+tf)$ (85.21).

In [33][32], the author made an experimental study to compare two approaches of reduction dimensionality and verifies their effectiveness in Arabic document classification. Firstly, he applied latent Dirichlet allocation (LDA) and latent semantic indexing (LSI) for modeling the corpus contained 20,000 documents. He generated two matrices LDA (documents/topics) and LSI (documents/topics). Then the SVM algorithm is used for document classification, which is known as an efficient method for text mining. Classification results are evaluated by precision, recall and F-measure. The experiment shows that the results of dimensionality reduction via LDA outperform LSI in Arabic topic classification.

Said in [27] presented an evaluation study of the benefits of using morphological tools in Text Classification. The study includes using the raw text, the stemmed text, and the root text. The stemmed and root text are obtained using two different preprocessing tools. The results show that using light stemmer combined with a good performing feature selection method such as mutual information enhances the performance of Arabic Text Categorization especially for small sized data sets and small threshold values. Additionally, using the raw text leads to the worst performance in small datasets while its performance was among the best tools in large datasets. This may explain the contradiction in the results obtained previously in the literature of the Arabic text categorization since the performance of the preprocessing tools is affected by the characteristics of the dataset used.

3 Discussion

From previous discussion, most of related work in the literature used small in-house collected corpus, and applied one or two classifiers to classify one corpus which is not enough to evaluate Arabic TC. Thus, there are contradictions between results of researches in the literature because of using different corpora and different preprocessing techniques.

In addition, the impact of text preprocessing and different term weighting schemes combinations on Arabic text classification using popular text classification algorithms has not been studied in the literature. Also, there is a debate among researchers about the benefits of using morphological tools in TC.

We summarized the research problems in the following points:

- Debate among researchers about the benefits of using English morphological tools in TC. To the best of our knowledge, the benefits of using Arabic morphological tools (stemming and light stemming) is not addressed for Arabic Language; only [14][19] applied on single corpus belong to only 3 categories.
- The impact of text preprocessing and different term weighting schemes combinations on Arabic text classification using popular text classification algorithms has

not been studied in the literature. Only [25][27] have addressed the impact of morphological analysis tools on Arabic text classification. Their work is not comprehensive regarding Arabic corpora, classifiers, and term weighting schemes.

- The lack of availability of publically free accessible Arabic Corpora.
- The lack of standard Arabic morphological analysis tools.
- Most of related works in the literature used small in-house collected corpus.
- Most of related works in the literature applied one or two classifiers to classify one corpus. This is not enough to evaluate Arabic TC.
- There are contradictions between results of researches in the literature because of using different corpora and different preprocessing techniques.

For this, we have focused as objectives to build the largest publically free accessible Arabic Corpora, implement and integrate Arabic morphological analysis tools, conduct a comprehensive study about the impact of text preprocessing on Arabic text classification, and develop a method of representation and indexing text reflecting more semantics.

We start by building our data set. The corpus contains 20.000 documents that vary in writing styles. These documents fall into 10 categories that equal in the number of documents. In this Arabic dataset, each document was saved in a separate file within the directory for the corresponding category, i.e., the documents in this dataset are single-labeled. Tables 2 and 3 show more specified details about the collection.

Table 2. Dataset

NB of text in the corpus	20.000
NB of words in the corpus	2 .523 .022
Size of corpus (Mb)	34.0 Mb
NB of category	10

Table 3. Number of documents in each category

OATC	NB of text	Average number of words per text	Number of words per category	Category Size (Mo)
Sport	2 000	141.261	282 522	2.99
regional	2 000	125.723	251 447	2.71
Culture	2 000	168.485	336 971	3.62
world	2 000	105.701	211 402	2.26
National	2 000	136.739	273 479	2.97
political	2 000	164.356	328 712	3.53
Economic	2 000	148.922	297 845	3.27
Student	2 000	203.485	406 971	4.50
Investigation	2 000	253.602	507 205	5.43
Judicial	2 000	126.93	253 860	2.70

The corpus are collected from online Arabic newspapers including <http://www.attounissia.com.tn>, www.alchourouk.com/, www.assabahnews.tn/, <http://jomhouria.com/>, Table 4 summarize the percentage of split between different sources. As we can show for example, the “sport” category is composed of 25% from Attounissia, 25% from Alchourouk, 25% from Assabahnews, 25% from Jomhouria.

Table 4. Percentage of split between different sources

Source	Attounissia	Alchourouk	Assabahnews	Jomhouria
Sport	25%	25%	25%	25%
Regional	-	50%	50%	-
Culture	25%	25%	25%	25%
word	25%	25%	25%	25%
National	25%	25%	25%	25%
Political	-	100%	-	-
Economic	50%	-	-	50%
Student	100%	-	-	-
Investigation	100%	-	-	-
Judicial Incidents	25%	25%	25%	25%

4 Conclusion

In this paper we have presented a brief current state of the Art for Arabic text representation and classification methods. All major problems in the most of related work in the literature used small in-house collected corpus, and applied one or two classifiers to classify one corpus which is not enough to evaluate Arabic TC. Thus, there are contradictions between results of researches in the literature because of using different corpora and different preprocessing techniques. In addition, the impact of text preprocessing and different term weighting schemes combinations on Arabic text classification using popular text classification algorithms has not been studied in the literature. Also, there is a debate among researchers about the benefits of using morphological tools in TC. For this, we have built the largest publically free accessible Arabic Corpora and in future work we will implement and integrate Arabic morphological analysis tools, conduct a comprehensive study about the impact of text preprocessing on Arabic text classification, and develop a method of representation and indexing text reflecting more semantics.

References

1. Saad, M. K. (2010). The impact of text preprocessing and term weighting on arabic text classification (Doctoral dissertation, The Islamic University-Gaza).
2. El-Kourdi M., Bensaïd A. and Rachidi T. (2004). Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. 20th International Conference on Computational Linguistics. August, Geneva. Pp51—58

3. El-Halees A. (2007), "Arabic Text Classification Using Maximum Entropy", *The Islamic University Journal (Series of Natural Studies and Engineering)*, 15(1), pp. 157-167.
4. Sawaf H., Zaplo J., Ney H. (2001), "Statistical Classification Methods for Arabic News Articles", In the Workshop on Arabic Natural Language Processing, ACL'01, Toulouse, France.
5. Al-Zoghby A., Eldin AS., Ismail NA., Hamza T. (2007), "Mining Arabic Text Using Soft Matching association rules", In the Int. Conf. on Computer Engineering & Systems, ICCES'07, pp 421 - 426.
6. Mesleh A. (2007), "Chi Square Feature Extraction Based SVMs Arabic Language Text Categorization System", *Journal of Computer Science*, 3(6), pp. 430-435.
7. Mesleh A. (2007), "Support Vector Machines based Arabic Language Text Classification System: Feature Selection Comparative Study", In the 12th WSEAS Int. Conf. on APPLIED MATHEMATICS, Cairo, Egypt.
8. Harrag F., El-Qawasmeh E., Pichappan P. (2009), "Improving Arabic text categorization using decision trees", In the 1st Int. Conf. of NDT '09, pp. 110 – 115.
9. Al-Shalabi R., Kannan G., Gharaibeh H. (2006), "Arabic text categorization using KNN algorithm", In the Proc. of Int. multi conf. on computer science and information technology CSIT06.
10. Khreisat L. (2009), "A machine learning approach for Arabic text classification using N-gram frequency statistics", *Journal of Informetrics, Elsevier*, 3(1), pp. 72-77.
11. Harrag F., El-Qawasmeh E. (2009), "Neural Network for Arabic text classification", In the 2nd Int. Conf. of Applications of Digital Information and Web Technologies, ICADIWT '09, pp. 778 – 783.
12. Hmeidi I., Hawashin B., El-Qawasmeh E. (2008), "Performance of KNN and SVM classifiers on full word Arabic articles", *Journal of Advanced Engineering Informatics* 22, pp. 106–111.
13. Abbas M., Smaili K., Berkani D. (2009), "Comparing TR-Classifer and KNN by using Reduced Sizes of Vocabularies", In The 3rd Int. Conf. on Arabic Language Processing, CITALA2009, Mohammadia School of Engineers, Rabat, Morocco.
14. Duwairi R. (2007), "Arabic text Categorization", In the Int. Arab journal of information technology, 4(2).
15. Kanaan G., Al-Shalabi R., Ghwanmeh S. (2009), "A comparison of text-classification techniques applied to Arabic text", *Journal of the American Society for Information Science and Technology*, 60(9), pp. 1836 – 1844,.
16. Al-Harbi S., Almuhareb A., Al-Thubaity A., Khorsheed M., Al-Rajeh A. (2008), "Automatic Arabic Text Classification", In JADT'08, France, pp. 77-83.
17. Bawaneh M., Alkoffash M., Al-Rabea A. (2008), "Arabic Text Classification using K-NN and Naïve Bayes", In *Journal of Computer Science*, 4 (7), pp. 600-605.
18. El-Halees A (2008), "A Comparative Study on Arabic Text Classification", *Egyptian Computer Science Journal* 20(2).
19. Duwairi R (2005), "A Distance-based Classifier for Arabic Text Categorization", In the Proc. of the Int. Conf. on Data Mining, Las Vegas, USA.
20. Duwairi R. (2006), "Machine Learning for Arabic text Categorization", *Journal of the American Society for Information Science and Technology*, 57(8), pp. 1005-1010.
21. Alruily M., Ayesh A, Zedan H. (2009), "Crime Type Document Classification from Arabic Corpus", In the 2nd Int. Conf. on Developments in eSystems Engineering, pp.153-159.
22. Abbas M., Smaili K., Berkani D. (2009), "A Trigger-based Classifier", In The 2nd Int. Conf. on Arabic Language Resources and Tools (MEDAR 2009), 22-23, Cairo, Egypt.

23. Ayadi, R., Maraoui, M. and Zrigui, M. (2011) 'SCAT: a system of classification for Arabic texts', *Int. J. Internet Technology and Secured Transactions*, Vol. 3, No. 1, pp.63–80.
24. Syiam M., Fayed Z., Habib M. (2006), "An Intelligent System for Arabic Text Categorization", In *IJICIS*, 6(1), pp. 1-19.
25. Duwairi R., Al-Refai M., Khasawneh N. (2009), "Feature reduction techniques for Arabic text categorization", *Journal of the American Society for Information Science*, 60(11), pp. 2347-2352.
26. Thabtah F., Hadi W. Musa, Al-shammare G. (2008), "VSMs with K-Nearest Neighbour to Categorize Arabic Text Data", in the Proc. of the World Congress on Engineering and Computer Science, WCECS'2008, San Francisco, USA.
27. Said D., Wanas N., Darwish N., Hegazy N. (2009), "A Study of Arabic Text preprocessing methods for Text Categorization", In the 2nd Int. conf. on Arabic Language Resources and Tools, Cairo, Egypt.
28. Alsaleem S. (2011), "Automated Arabic Text Categorization Using SVM and NB", in the *International Arab Journal of e-Technology*, Vol. 2, No. 2, pp. 124-128.
29. Majed Ismail Hussien, Fekry Olayah, Minwer AL-dwan, Ahlam Shamsan (2011), "ARABIC TEXT CLASSIFICATION USING SMO, NAÏVE BAYESIAN, J48 ALGORITHMS", in the *IJRRAS* 9 (2), pp. 306-316.
30. Maraoui, M., Antoniadis, G., & Zrigui, M. (2009, July). SALA: Call System for Arabic Based on NLP Tools. In *IC-AI* (pp. 168-172).
31. Zouaghi, A., Zrigui, M., & Antoniadis, G. (2008). Automatic Understanding of Spontaneous Arabic Speech-A Numerical Model. *TAL*, 49(1), 141-166.
32. Ayadi, R., Maraoui, M., & Zrigui, M. (2014). Latent Topic Model for Indexing Arabic Documents. *International Journal of Information Retrieval Research (IJIRR)*, 4(1), 29-45.
33. Ayadi, R., Maraoui, M., & Zrigui, M. (2015, October). LDA and LSI as a Dimensionality Reduction Method in Arabic Document Classification. In *International Conference on Information and Software Technologies* (pp. 491-502). Springer International Publishing.

A Word Embeddings Model for Sentence Similarity

Victor Mijangos, Gerardo Sierra and Abel Herrera

National Autonomous University of Mexico
Language Engineering Group, Faculty of Engineering
Mexico City, Mexico
{vmijangosc,gsierram}@iingen.unam.mx, abelherrera1@gmail.com

Abstract. Currently, word embeddings (Bengio et al, 2003; Mikolov et al, 2013) have had a major boom due to its performance in different Natural Language Processing tasks. This technique has overpassed many conventional methods in the literature. From the obtained embedding vectors, we can make a good grouping of words and surface elements. It is common to represent top-level elements such as sentences, using the idea of composition (Baroni et al, 2014) through vectors sum, vectors product or through defining a linear operator representing the composition. Here, we propose the representation of sentences through a matrix containing the word embedding vectors of such sentence. However, this involves obtaining a distance between matrices. To solve this, we use a Frobenius inner product. We show that this sentence representation overtakes traditional composition methods.

1 Introduction

Word embeddings methods, based on the proposal of neural language models [2], have over-passed traditional approaches to natural language processing in a lot of tasks [10], [3]. The distributional space model that word embeddings proposes seems to codify better the proper features of the language. In general terms, the distributional models have been used for representing words in a vector space. Nevertheless, most of the time we look for representing high-level linguistic elements. For example, here we want to find for sentences similarity through its representation on a vector space.

For representing such elements, [1] summarizes three compositional methods that can allow us to represent high-level linguistic terms:

1. Vectors sum.
2. Vectors multiplication.
3. A linear operator through representation of the compositionality of a word.

Vectors sum and vectors multiplication are easy to implement. But to determine a linear operator representing the compositional of a word becomes a hard labor. The problem with these compositional approaches is that they assume

that language is compositional. Given this, we can find many counterexamples. Therefore, we believe that a perspective that compare the similarity between the words that make a sentence, rather than the sentences themselves, can give good results in determining similarity between sentences.

2 Previous works

Similarity between sentences has been an essential problem for text mining, question answering, text summarization and another tasks. What we want is that given two sentences, we can determine their similarity through semantic proximity. Nevertheless, this task shows big complications. As we have mentioned before, most of the time the assumption of compositionality is given. So, most of the methods are based on this approach to represent linguistic items in vector spaces [1], [7].

Some of the methods that have been used to determine similarity measures for sentences are compared in (Achananuparp, 2008). One of these methodologies is the word overlap measure. This method seeks to find related concepts through comparing the overlapping words between definitions (sentences) of both concepts. In (Metzler, 2005), a method of word overlap is evaluated from a simple word overlap fraction that determines the proportion of words that appear in the two sentences to compare them. Then this proportion is normalized through the length of the sentences. Another method based on word overlap used to determine similarity between sentences is the Inverse Document Frequency (IDF) overlap, where the proportion of words is compared in two sentences from their IDF weights.

Based on these ideas, (Banerjee et al., 2003) extends the concept of word overlap to the distinction between multi-word terms. They assume that the simple word overlap methods do not take into account the elements that are composed of more than one word and that may be important for the similarity between sentences. Therefore, they estimate the overlap between multi-word terms and single word terms.

Other methods are based on Term Frequency-Inverse Document Frequency (TFIDF). For example, (Allan, 2003) is based on search thematically similar sentences. It is based on the sum of the TFIDF values of the words in both sentences, since it is assumed that this measure weights the thematically relevant words. Another way is to create vectors based on the idea of Bag of Words (BoW) and compare the distance between vectors sentences. However, BoW lose relevant information between the sentences, such as order, and have now been overtaken by the word embeddings representations.

In (Landauer, 1997) it is proposed that the similarity between sentences is given from a linear combination of vectors of semantic similarity and the similarity in words order.

Meanwhile, in (Achananuparp, 2008a) a methodology of similarity between sentences for question-answering systems is proposed. They propose an hybrid approach, combining semantic similarity and syntactic patterns; for this, they

take into account the similarity between words, word order and Part of Speech tags. Also, they integrate a Support Vector Machine to determine information about the types of sentences.

As reported by Achananuparp (2008), the method with the highest accuracy is proposed by (Achananuparp, 2008a), followed by other methods that rely in similar concepts. However, such methods require a high cost, since they are based on principles of word order, labeling Parts of Speech and semantic similarity. Meanwhile, the methods of word overlap are generally not as efficient, as they do not capture the paraphrase or syntactic level elements, rather remaining at the lexical level.

Socher (2011) proposes the analysis of similarity between paraphrases through a recursive auto-encoder (RAE). They represent the sentence as an ordered list of word vectors, $s = \{x_1, \dots, x_m\}$ obtained with a word embedding algorithm. Given a sentence s the binary parse tree has the form of branching triplets ($p \mapsto c_1 c_2$), where p is a partner and c_i is a children, such that c_i is a word vector or a non-terminal node. Then $p = f(W_e[c_i; c_j] + b)$, where $[c_i; c_j]$ is the concatenation of the corresponding word vectors and W_e is the encoding matrix to learn.

The error function is given by:

$$E_{rec}(p) = || [c_1; c_2] - [c'_1; c'_2] ||^2 \quad (1)$$

where $[c'_1; c'_2] = f(W_d p + b)$. This process is recursively applied until the tree is fully constructed. To assign a similarity between two sentences, a euclidean distance is computed between all words and phrases vectors. This euclidean distance fills a similarity matrix. To determine a similarity between paraphrases, the authors use a pooled matrix. Nevertheless, as the authors said, the pooled matrix loses some of the information of the original matrix.

Another approximation to sentence similarity is showed by Kartsaklis (2012). They propose a tensor-based method. They create tensors through grammatical information and Frobenius algebras. Nevertheless, this method requires the grammatical information and, also, does not use the matrix representation for computing the similarity but a \mathbb{R}^n representation.

Based on this ideas, Kim (2015) proposes a tensor-based composition for the word embedding algorithm. They used the compositional methods explained in (Milajevs, 2014) for integrating the compositional function into the neural network of the word embeddings. They proved the typical mean and point-wise multiplication techniques; additionally, they proved the sum and the concatenation of these two techniques. Furthermore, they proposed tensor-based techniques; nevertheless, they used a projection function to obtain a vector in \mathbb{R}^n from the matrix representation.

A comparison of different methods is made by Milajevs (2014). They show the simple vector addition and multiplication methods as well as the Frobenius algebras based methods. However, for the computation of similarity between sentences they use a vector representation in all cases combined with a cosine distance.

Our proposal seeks to be more simple, and at the same time to capture elements that are beyond the lexical level. At the same time, the method proposed

here does not lose information of the similarity between words. This proposal seeks to characterize a sentence from the words that compose it, and to determine a matrix space where a geometry can be defined through a Frobenius inner product. Thus, the method proposed here becomes simple compared with the categorical compositional distributional and projection methods. Furthermore, it has good performance.

3 Theoretical framework

The methodology proposed here consists of two main tools: word embeddings and Frobenius inner product. In the following section we describe these tools.

3.1 Word embeddings

In the area of Computational Linguistics, it is common to hear about language models. These language models can be seen as Markov process of order r . Overall, Markov models allow us to see the likelihood of a chain as the product of the states that make up the chain. The language models are simple models that attempt to determine the likelihood of a linguistic element from the n above items. Thus we can understand them as we show below.

Definition 1 (n-grams models) *Giving a string of words $w_{1,n} = \{w_1, \dots, w_t\}$ the language model determines the probability of the chain from the probability of the states of the chain. Such that:*

$$P(w_{1,t}) = \prod_{i=1}^t P(w_i | w_{i-n+1} \dots w_{i-1}) \quad (2)$$

where $n \in \mathbb{N}$ is the length of the window.

Perhaps the simplest model within these is the bi-gram model. It takes $r = 2$ such that we have a simple Markov process, then equation (2) becomes:

$$P(w_{1,t}) \approx \prod_{i=1}^t P(w_i | w_{i-1}) \quad (3)$$

Based on these models, [2] proposes a model to determine the probability of a linguistic element given its context. To do this, it is proposed the use of deep learning. Therefore, these new models are called ‘neural language model’. As [2] noted, this idea is based on three main points:

1. Representing each word in a vocabulary through a distributed feature vector (a vector with entries in \mathbb{R}).
2. Expressing a joint probability function of words sequences in terms of the features vectors of the words in the sequence.

3. To learn simultaneously the feature vectors and the parameters of the probability function.

The probability function, in this case, is of the type expressed in equation (2). Now, however, a neural network is used to predict subsequent words in the string. In this case, the vectors representing the word are learnt by a learning machine. In general, the model is based on the ideas of distributional models; the vectors of similar words should be similar, and the context of words plays an important role.

We have a vocabulary $\mathfrak{C} = \{w_1, \dots, w_t\}$ of finite length. We want to learn a model $f(w_{i-n+1} \dots w_i) = P(w_i | w_{i-n+1} \dots w_i)$ such that $\sum_{j=1}^t f_j(w_{i-n+1} \dots w_{i-1}) = 1$; this means that we look for f to be a probability measure. We can define f as:

$$f_j(w_{i-n+1} \dots w_{i-1}) = g_j(v(w_{i-n+1}) \dots v(w_{i-1})) \quad (4)$$

where

- $v : \mathfrak{C} \rightarrow \mathbb{R}^m$ is the function determining a vector in \mathbb{R}^m for each word in the vocabulary. In general terms, $v(\cdot)$ is represented as a matrix of size $|\mathfrak{C}| \times m$.
- A function g (a neural network) that maps a sequence of input vectors, taking into account its context, to a conditional probability distribution for the next word w_i . The output vector generated by g is a probability vector such that the i th entry estimates the probability of $P(w_i | w_{i-n+1} \dots w_{i-1})$.

To determine the probability of each word given the $n + 1$ previous words, the Softmax regression is used. The Softmax regression is a form of asymmetric probability of neighbors (Hinton, 2002) where $\delta = \langle \cdot, \cdot \rangle$ is the inner product, such that:

$$P(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\exp(\langle v_i, h(v_{i-n+1} \dots v_{i-1}) \rangle)}{\sum_{k=1}^t \exp(\langle v_k, h(v_{i-n+1} \dots v_{i-1}) \rangle)} \quad (5)$$

where $v_k = v(w_k) \forall k \in \{1, \dots, t\}$ and $h : \mathbb{R}^{m \times (n+1)} \rightarrow \mathbb{R}^m$ is a function that maps the word vectors of the context to a sole vector in \mathbb{R}^m .

Therefore, the model proposed in (2) now depends on the equation (5). However, what we want is a vector space representation of sentences.

In general terms, the method of word embeddings is based on stochastic neighbor embedding, watching the elements in the context of a word as neighbors. Therefore it seeks to maximize the equation (5) while a loss function, that often tends to be the Kullback-Liebler divergence (Hinton, 2002), is minimized. So we want to minimize:

$$KL(p||q) = \sum_{i=1}^t p_i \log \frac{p_i}{q_i} \quad (6)$$

However, since the Kullback-Leibler divergences represents a high computational cost, the use of negative sampling has been proposed [6], which streamlines the process of word embeddings.

Then we have a word w and its context, $N(w)$. To generate the vectors of dimension m , where each coordinate represents a feature that is learnt by a neural network, we follow the next steps:

1. We randomly generated a matrix $V \in \mathbb{R}^{t \times m}$ of input vectors and a different matrix $V' \in \mathbb{R}^{m \times t}$, where t is the number of words.
2. Given the context, we determine $P(w_i, N(w_i))$ with softmax regression. We choose the function h as follows:

$$h := \frac{1}{n} \sum_{k=1}^n v'_k \quad (7)$$

where $v' = v'(w)$ is the word vector of the matrix $(V')^t$.

3. We upgrade from a hidden layer to the output of the neural network, by using a stochastic gradient descent with a learning rate η , such that:

$$v'_{i+1}(w_j) = v'_i(w_j) - \eta \nabla \epsilon(w_j) \quad (8)$$

where ϵ is the loss function. We use a negative sampling such that we define the loss function as follows.

$$\nabla \epsilon(w_j) = \begin{cases} 1 - P(w_j|N(w_j))h(N(w_j)) & \text{if } w_j = w_o \\ 0 - P(w_j|N(w_j))h(N(w_j)) & \text{if } w_j \neq w_o \end{cases} \quad (9)$$

where w_o is the actual objective word in the iteration.

4. To upgrade the input to the hidden layers, given $e_k \in N(w_j)$ we have $\forall m' \in 1, \dots, m$

$$v_{i+1}(w_k) = v_i(w_k) - \eta \sum_{j=1}^n \epsilon(w_j) \cdot v'_{m',j} \quad (10)$$

where $v'_{m',j}$ represents the m' th entry of the matrix V' .

The algorithm then iteratively runs until the cost function is less than a given range. Also, it can be run a determined number of iterations.

This way, what the algorithm does is try to approximate the distributions between the matrices V and $(V')^t$ from the observations of the contexts in which a word occurs.

Currently, the methods based on word embeddings have surpassed previous models in most of the tasks of natural language processing. Also they have the advantage of not requiring a dimensionality reduction because the dimension is chosen a priori. Therefore we work with vectors having low dimensionality unlike other methods. It is important to point out that the distributional assumptions [5], [9], are still presented, implying that word embeddings are capturing the similarity of the words from the idea that similar words appears in similar context (Goldberg, 2014).

The method of word embeddings generates word vectors. From these word vectors we can perform a composition process, which commonly consists of the

sum vector (it has also been used despite other methods described above). However, what we propose is to characterize a sentence from a matrix where each row vector is a vector representation of the words that compose such a sentence. However, before this, we are faced with reinventing the classic methods of distance between vectors. For this, we use the Frobenius inner product and related concepts explained below.

3.2 Frobenius distance

To calculate the distance between two matrices, we first have to define a geometry in the space of matrices of $m \times n$. For this, we require an inner product. That is, a function that, given u, v , and w vectors and a scalar $\lambda \in \mathbb{R}$, the next properties are met:

- Positive-definite** $\langle v, v \rangle \geq 0$.
- Non-degenerative** $\langle v, v \rangle = 0$ iff $v = 0$.
- Bilinear** $\langle \lambda u + w, v \rangle = \lambda \langle u, v \rangle + \langle w, v \rangle$.
- Conjugated symmetry** $\langle u, v \rangle = \overline{\langle v, u \rangle}$.

Given this, the following theorem gives us an inner product into the space of matrices of $m \times n$ [4], [8].

Theorem 1 *The equation*

$$\langle X, Y \rangle = \text{trace}(X^t \cdot Y) = \sum_{i=1}^n \langle X_i, Y^i \rangle \quad (11)$$

defines a inner product in the spaces of $\mathbb{R}^{m \times n}$.

It is easy to see that given the properties of the trace and the inner product over the column vectors of a matrix, the equation (11) is positive-definite. We can also see that it is bilinear, as we have:

$$\begin{aligned} \langle \lambda X + Z, Y \rangle &= \text{trace}((\lambda X + Z)^t \cdot Y) \\ &= \sum_{i=1}^n \langle \lambda X_i + Z_i, Y^i \rangle \\ &= \lambda \langle X_i, Y^i \rangle + \langle Z_i, Y^i \rangle \\ &= \lambda \langle X, Y \rangle + \langle Z, Y \rangle \end{aligned}$$

Finally, it satisfies the property of conjugated symmetry as we have:

$$\begin{aligned} \langle X, Y \rangle &= \sum_{i=1}^n \langle X_i, Y^i \rangle \\ &= \sum_{i=1}^n \langle (Y^i)^t, (X_i)^t \rangle \\ &= \langle Y, X \rangle \end{aligned}$$

Since equation 11 indeed defines an inner product, then we can move on to define a norm on the space of matrices of $m \times n$. This norm is known as the Frobenius norm and is defined as (see (Tarazaga, 2001)):

$$\|A\| = \sqrt{\langle A, A \rangle} = \sqrt{\text{trace}(A \cdot A^t)} \quad (12)$$

The norm defines a metric given by $\|A - B\|$ where A and B are matrices. Nonetheless, we seek a way to standardize this distance. Given the properties of the norm and the inner product, it is clear that the Cauchy-Schwartz inequality is satisfied and therefore we can define a distance function as follows:

$$\cos(A, B) = \frac{\langle A, B \rangle}{\|A\| \cdot \|B\|} \quad (13)$$

$$= \frac{\text{trace}(A, B^t)}{\sqrt{\text{trace}(A \cdot A^t) \cdot \text{trace}(B \cdot B^t)}} \quad (14)$$

So the distance between two matrices holds that $|\cos(A, B)| \leq 1$. So we now have a normalized distance between matrices that can be applied to the matrices generated from the vectors of the words composing each sentence.

4 Proposal

Based on what has been showed, we propose to calculate the similarity of two sentences from their matrices created by embedding vectors of words and the use of a Frobenius based distance to determine the similarity between the two matrices in the space. We start with the following elements:

- $Q = \{s_1, \dots, s_m\}$ a set of sentences.
- $W = \{w_1, \dots, w_k\}$ a set of words obtained from the sentences in Q .

Now, we choose a sentence set, this set will be the training set, then the words are split in order to obtain the vector representation used to generate the matrix. However, a large sample number is required to obtain word embeddings. In consequence, we used the Corpus del Español Mexicano Contemporáneo (CEMC) (DE, 1987s) that has around two millions of words.

So, the set of words W is done from CEMC corpus. It allows that a sentence not in Q can be compared even if its words are not in the training set of sentences.

The proposal consists of five steps: words representation in a vector space, sentences representation, petition representation and the compute of similarity between sentences. These steps will be explained below:

Word representation. A vector space of dimension n is generated (the dimension is 800) from the CEMC using word embeddings. We use word2vec [7] [6]. The window equals to 5 and the minimum occurrence is 1. With these parameters all the words are included and the algorithm has more range when we ask for a query. In this step, the vector space $W \subset \mathbb{R}^m$ is obtained, each vector is the representation of one word of CEMC.

Sentence Representation. The sentence representation in Q is done from matrices where each line is one vector of a word in W . To do this, first of all, the sentence is divided in its words, where a vector $v \in W$ is the representation of the word. So, the method obtains a matrix of $r \times n$, where n is the dimension of W (the chosen dimension is equals to 800) and r is the number of the sentence words. Finally, the method obtains a set $S = \{s_1, \dots, s_m\}$ where each $s_i, i = 1, \dots, m$ is the matrix representation of the sentence i .

Petition Representation. The petition is a sentence; the method will look for the Q element most similar. The petition is called q_0 , not necessarily in Q . It is replaced by the vectors representation of its words, so it forms a matrix s , of $l \times n$ dimension.

Sentence similarity. Finally, the objective is to determine a distance between s' , the matrix representation of q_0 , and all the elements of S . This function will give us the most similar sentence to s' . The reader can notice that the matrices are not of the same dimension but have the same number of columns. In the method we propose the use of the Frobenius inner product to compute this distance. So, the similarity function between the matrices is given by equation 13 and equation 11. Then the sentence most similar to q_0 is given by

$$\arg \max_i \delta(s', s_i), s_i \in S \tag{15}$$

where δ is the distance between the matrices.

The process is described at Figure 1.

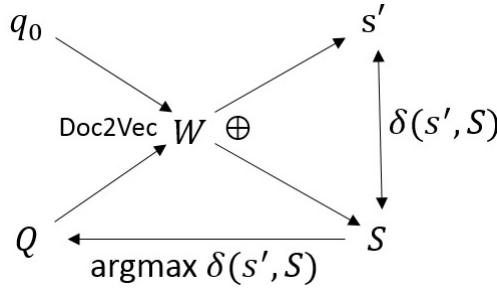


Fig. 1. Process to compute the similarity between sentences.

The advantages of our proposal over other methods reside in the computation of the similarity through the matrix representation. We do not need to make any map into \mathbb{R}^n for computing a metric between sentences. The Frobenius inner product allows us to determine a primal geometry over the space of matrices. With this geometry we can calculate the similarity between points in this vector space (these points are matrices). But what happens in a deep level is that the Frobenius inner product computes a standard inner product over the row vectors

representing the words of the sentences. Then, the Frobenius inner product can be seen as follows:

$$\langle A, B \rangle = \sum_{i=1}^r \langle a_i, b_i \rangle \quad (16)$$

where a_i is the i th word vector composing the sentence represented by the matrix A and b_i the i th word vector of the sentence represented by the matrix B .

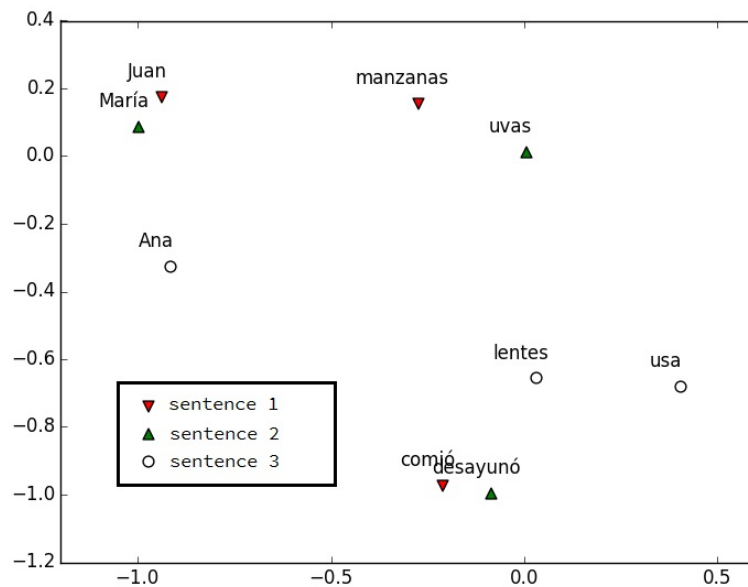


Fig. 2. Plot of the words composing the similar sentences ‘Juan comió manzanas’ (‘John ate apples’) and ‘María desayuna uvas’ (‘Mary had grapes for breakfast’), and the dissimilar sentence ‘Ana usa lentes’ (‘Ana wears glasses’).

Take a sentence like ‘Juan comió manzanas’ (‘John ate apples’). This sentence is similar, for one hand, to ‘María desayunó uvas’ (‘Mary had grapes for breakfast’) since the verbs ‘comió’ y ‘desayunó’ are similar; both verbs reflect the action of ‘eat’, past tense and third singular person. Also, the direct objects ‘uvas’ and ‘manzanas’ are similar because they both are fruits. For the other hand, a sentence like ‘Ana usa lentes’ (‘Ana wears glasses’) has a completely different meaning. Here, the verbs ‘wears’ and ‘ate’ are from different semantic groups, such as ‘glasses’ and ‘apples’. If we plot the three sentences we can

see that the words composing them are grouped differently depending on its meaning (see Figure 2). Computing the Frobenius cosine between the two similar sentences ‘Juan comió manzanas’ and ‘María desayunó uvas’ we have 0.62. Whereas for ‘Juan comió manzanas’ and ‘Ana usa lentes’ the Frobenius cosine equals 0.39.

Our method tries to make the calculation of similarity between sentence simpler than than other methods like the proposed in (Milajevs, 2014) or (Socher, 2011) by computing the metric in the original matrix space. This implies that there is no information loss and we do not need to map the representation into another simpler vector space (generally \mathbb{R}^n).

5 Results

For evaluation, a corpus of Spanish paraphrases was used; this corpus has 144 sentences with paraphrases. The corpus always has an original document and two paraphrase levels: low-level paraphrases and upper-level paraphrases. The low-level paraphrases has lexical and syntactic changes and, in general, formal level changes. The upper-level paraphrases has changes at discursive level, semantic and more elaborated changes. Then, for the evaluation process, the method only uses low level paraphrases, and they are compared to the original texts. More information of this corpus is at (Mota).

The baseline is the paraphrase representation from word vectors sum, the vectors were obtained from the words embedding method explained above. The cosine and inner product are compared at both cases, then the accuracy is obtained. Using the cosine distance, the highest accuracy is achieved by our method. Unfortunately, only by a difference of 0.03; this improvement is not significant. However, with the inner product, our method achieve a 0.62 accuracy, against the accuracy of 0.55 obtained by the vectors sum method (see table 1).

Table 1. Results comparison.

Method	Inner product	Cosine
Matrix representation	0.6206	0.5862
\mathbb{R}^n representation	0.0689	0.5517

6 Conclusions and Future work

The proposed method has a better performance compared to the traditional \mathbb{R}^n vector representation. The \mathbb{R}^n representation method is commonly used to represent linguistic elements; now, the information from that method can be amplified using a matrix representation. This way, the a priori conception of compositionality is not assumed. In the matrix representation method, each word represents

a column of a matrix and we can compare each word with the words of another sentence. So, a matrix representation is more adequate than the representation through vectors in \mathbb{R}^n .

The advantage over other methods, such as the one proposed in (Socher, 2011) is that our proposal does not lose information. The Frobenius inner product is a natural geometric form in the matrix space. This inner product compares the distance between all the words of both sentences to give us a similarity measure of the sentences.

However, a lot of work must be done. The matrix representation method must be compared with other methods. In this paper we prove this method for Spanish language. Nevertheless, a more complete set of languages can show that this is a good language-independent method.

We want to point out that, as the angle within matrices makes no much sense, the Frobenius inner product improve the result of the experiment. Nevertheless, we use just two distances to compare. For later works, it is necessary to define metrics that capture better the topology of the data. Also, the word embeddings methods need big size corpora. So a bigger corpus than CEMC is needed for a question-answering system application.

The sentence representation can be improved. We see that, by the definition of the Frobenius inner product, the word order between the sentences improves the results. This is given because it is expected that both sentence were composed by the same syntactic structure. Nevertheless, this happens rarely. Other elements as syntactic information and morpho-syntactic information can be integrated. The contribution of these elements must be proved.

References

1. Baroni, M., Bernardi, R., Zamparelli, R.: Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology* 9 (2014)
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. *The Journal of Machine Learning Research* 3, 1137–1155 (2003)
3. Campr, M., Ježek, K.: Comparing semantic models for evaluating automatic document summarization. In: *Text, Speech, and Dialogue*. pp. 252–260. Springer (2015)
4. Chang, K.C., Pearson, K., Zhang, T., et al.: Perron-frobenius theorem for nonnegative tensors. *Commun. Math. Sci* 6(2), 507–520 (2008)
5. Harris, Z.: Distributional structure. *Papers on Syntax* pp. 3–22 (1954)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)
8. Noutsos, D.: On perron–frobenius property of matrices having some negative entries. *Linear Algebra and its Applications* 412(2), 132–153 (2006)
9. Sahlgren, M.: The distributional hypothesis. *Italian Journal of Linguistics* 20, 33–54 (2008)
10. Sateli, B., Witte, R.: Supporting wiki users with natural language processing. *Proceedings of the 8th Annual International Symposium on Wikis and Open Collaboration* pp. 379–423 (2012)

Using Multiple Metrics in Automatically Building Turkish Paraphrase Corpus

Bahar Karaođlan¹, Tarık Kışla², Senem Kumova Metin³,
Ufuk Hürriyetođlu¹, Katira Soleymanzadeh¹

¹Ege University, International Computer Institute, İzmir, Turkey
bahar.karaoglan@ege.edu.tr, {ufuk.hurriyetoglu,
katira.sole}@gmail.com

²Ege University, Department of Computer Education and Instructional Technology
tarik.kisla@ege.edu.tr

³İzmir University of Economics, Department of Software Engineering
senem.kumova@ieu.edu.tr

Abstract. Paraphrasing is expressing similar meanings with different words in different order. In this sense it is viewed as translation in the same language. It is an important issue in natural language processing for automatic machine translation, question answering, text summarization and language generation. Studies in paraphrasing can be classified as paraphrase extraction, paraphrase generation, paraphrase recognition. In this paper we present automatic sentential paraphrase extraction from comparable texts downloaded from Turkish newspapers related to similar news. We applied seven text similarity metrics and assumed the two most similar ones as candidates. Through an interface these are shown to 3 human annotators to be labelled as paraphrase, entailing, entailed, opposite in meaning and not paraphrase. In this paper we only present results driven from a single topic. The sentences in the other topics will be processed based on the experience gained in the current work. This will be the first automatically built and golden standard tagged Turkish paraphrase corpus.

Keywords: Paraphrase extraction, Turkish paraphrase corpus

1 Introduction

Identification of paraphrasing is an important issue in natural language understanding and information retrieval. As the first requirement golden standard tagged corpus for the assessment purposes is needed. Paraphrase identification is not much studied in Turkish and there is no corpus developed for Turkish paraphrase identification. In this paper we present an incremental methodology for selecting candidate paraphrase sentences for the human annotators. Basic idea is using different text similarity metrics to measure the similarity of each sentence to all the other sentences within the same topic. Then, take the two most similar sentences obtained from each metric as the candidates to be shown to annotators.

To create a paraphrase corpus different types of sources can be used to extract data. Source of data to be selected depends on the granularity of paraphrasing which can be at the phrase [1] [2] [3] sentence [4] [5] [6] [7] or paragraph level [8] [9]. Most of the studies have been at phrase level for automatic machine translation, information retrieval and information extraction purposes or sentence level for question answering, text summary and such. Comparable texts (Newspaper articles for the same news from different sources), parallel texts (bilingual or monolingual translations of the same text, answers given to the same question (FAQ, exams, customer opinions, rephrasing), text modifications (Wikipedia) are some of the data sources.

Once the data is collected and pre-processed, the next task is to tag the text pairs as paraphrase by the human annotators to achieve golden standard or training purposes. To ease the task of the annotators, text similarity techniques are exploited to choose the candidates automatically. These candidates are then marked as paraphrase or not paraphrase by the annotators.

Paraphrasing is a vague concept by itself and its understanding may vary from person to person. Some definitions [10] are as follows:

Wikipedia:	A restatement of a text or passage using different words
WordNet:	Express the same message in different words, rewording for the purpose of clarification
Purdue's OWL:	Your own rendition of essential information and ideas expressed by someone else, presented in a new form.
Pearson's glossary:	To record someone else's words in the writer's own words

To achieve a standard to some extent, guidelines are set for the annotation [5]. Even then, the boundary is not clear. In some studies paraphrases are labelled with degree of confidence in a graded fashion. For example in STS [11] and ULPC [7] textual similarity is annotated on 6 scale from exact semantic equivalence to complete un-relatedness. The annotation task is either done through field experts [7] or through crowdsourcing [11].

Along with paraphrasing studies, some researchers [12] have made distinction between paraphrasing and entailment in the labelling. Paraphrasing is interpreted as a bi-directional relation where the same meaning is derived from both texts. Textual entailment is interpreted as a directional relation where one text can be inferred from the other, but the reverse is not true. "Precise paraphrase" is addressed if the relation is bidirectional: Text A is a paraphrase of text B if and only if A entails B and B entails A [13].

In this study the sentences are annotated on five scale as 1) Paraphrase, 2) Entails, 3) Entailed, 4) Opposite and 5) Not paraphrase. We considered the first 3 as paraphrase and the last two as not paraphrase. Sentence 5, given as example below, is taken from our news database. Sentence 79 is labelled as paraphrase and sentence 1 is marked as entailed. Both sentences 5 and 79 give approximately the same message. Sentence 1 can be deduced from these sentences.

Sentence 05:

In the implementation, it is foreseen that shareholders use credit approximately 115% of their accretion that they get in four instalments up till now.

(Turkish: Uygulamada, hak sahiplerinin şimdiye kadar dört taksitte aldıkları nemanın yaklaşık yüzde 115'i kadar kredinin kullandırılması öngörüldü)

Sentence 79 (Paraphrase):

Credit will be given up to 115% of the total accretion that is received in 4 instalments till today.

(Turkish: Bugüne kadar 4 taksitte alınan toplam nemanın yüzde 115'ine kadar kredi verilecek)

Sentence 01 (Entailed):

To those who wish, Ziraat Bank will advance money on their receivable accretion.

(Turkish: Ziraat Bankası, isteyenlere alacakları nemalar karşılığında kredi kullandıracak.)

The assessment of the most linguistic studies heavily depends on the tagged corpus on which it is carried on. To increase the reliability of the golden standard, annotation is done by several experts and inter annotator agreement is calculated. The performance of paraphrase recognition approach is assessed with regard to the extent by which it correlates with human annotators as human annotators correlate with each other. For paraphrase studies, even in the cases where annotation rules are rigid, as in MSRP, high inter-annotator agreements cannot be achieved.

2 Related Work

Androutsopoulos and Malakasiotis [14] classify the studies in paraphrasing field with respect to two dimensions: 1) whether paraphrasing or entailment, 2) processing of paraphrases: generation, extraction or recognition. Since the main goal of this study is to create a Turkish gold-standard paraphrase corpus, we will focus on methodologies for paraphrase extraction. Paraphrase corpora have been developed for different purposes from different sources. We look at these studies from aspects given in Table 1.

Table 1. Classification of paraphrase corpora

Aspects	Explanation
Source	Comparable corpora (eg. News about the same event); bilingual corpora (using one language as a pivot to find paraphrases in other documents.), monolingual corpora (parallel translations of the same source), users (question answers, rephrases, twitter, Wikipedia, etc.)
Annotation	Automatic, experts, crowdsourcing
Granularity	Paraphrase, sentence, paragraph
Recognition basis	Syntactic similarity, semantic similarity, text alignment, word overlap
Rating	Binary, scaled
Purpose	Information retrieval, Automatic machine translation, Language generation, Question answering, Summarization

Microsoft Research Paraphrase corpus (MSRP) can be considered as the first major public paraphrase corpus [5] [15] annotated by humans in binary mode as paraphrase or not paraphrase. Two methods: string similarity measure and discourse-based heuristic are used to draw candidate sentences from news after applying support vector machine classifier. It consists of 5801 pairs of sentences, of which 67% are judged to be paraphrases.

User Language Paraphrase Corpus (ULPC) [7] is composed of 1998 sentence pairs taken from students rephrases in response to target sentences. To describe the quality of user response, 10 dimensions (garbage, frozen expression, irrelevant, elaboration, writing quality, semantic similarity, lexical similarity, entailment, syntactic similarity, paraphrase quality) of paraphrasing is considered. The annotators were asked to rate between 1-6 interval (1: minimum, 6: maximum) with equal distance that is, 1 and 6 denote negative or positive with maximum confidence, whereas 3 and 4 denote negative or positive with minimum confidence. The main purpose in posing this challenge was to facilitate intelligent tutoring systems to provide users with feedback comparable to those of experts.

Barzilay and McKeown [16] is an example to monolingual technique for corpus construction for the purpose of paraphrase extraction where, multiple English translations of the same literary text are used. Sub-sentential paraphrases were labelled as true and false by human annotators. They achieved 69% of accuracy in extracting paraphrases.

Two corpora in different languages are used to extract paraphrases in bilingual approach, taking one language as the pivoting language. Translations for phrases in the targeted language are found in the pivoting language using statistical and automatic machine translation techniques. Then, going backwards, translations for each of these in the targeted language are assumed to be paraphrase candidates. Colin and Callison-Burch [17] used the German-English the French-English, Spanish-English, and Italian-English portions of the Europarl corpus as sources. They report an accuracy of paraphrases extracted over multiple corpora as 57.4%.

Regneri et. al extracted paraphrase fragments from paraphrase sentences. With the aim of generality, they used sentential paraphrases from four different corpora: The Microsoft Paraphrase Corpus (MSR) [18] , The Microsoft Video Description Corpus (MSVD) [19], The TACoS Corpus [20], The “House” Corpus [21]. Two annotators labelled each ordered fragment pair as paraphrases, containment, backwards containment, unrelated or invalid. The overall annotator agreement was 0.50, according to Cohen’s Kappa (moderate agreement). Conflicts were resolved by a third annotator.

Agirre et al.’s [11] STS (The Semantic Textual Similarity) CORE corpus contains 2,250 pairs of headlines, machine translation evaluation sentences, and glosses (concept definitions). This corpus was annotated through crowdsourcing on 6-value scale as: 5: identical, 4: strongly related, 3: related, 2: somewhat related, 1: unrelated, 0: completely unrelated.

Bernhard et al. [22] developed QP (The Question Paraphrase Corpus) with the purpose of better understanding of how questions in social Q&A sites can be automatically analyzed and retrieved. 1000 questions and their paraphrases (in total 7434) are collected from randomly selected FAQ files in the Education category of the WikiAnswers web site. They report 80% accuracy for the task of question paraphrase retrieval.

3 Methodology

Our dataset is driven from Turkish BilCon2005 [23] news corpus which was created for the purpose of event detection and tracking. This corpus contains 209.305 news collected from five different Turkish news web sources: CNN Türk (<http://www.cnnturk.com>), Haber 7 (<http://www.haber7.com>), Milliyet Gazetesi (<http://www.milliyet.com.tr>), TRT (<http://www.trt.net.tr>), Zaman Gazetesi (<http://www.zaman.com.tr>) throughout the year 2005. 5872 of these news are profiled with Topic Title, Event Summary, What, Who, When, Where and other relevant data. This news is then categorized into 13 topics like natural disasters, accidents, bank, elections, and etc.

For the preliminary study we chose randomly the topic “Bank” and parsed it into sentences. After removing all the duplicates and short sentences with less than 3 words, 399 sentences are left. The average length of the sentences is 17.21 words, with the shortest 3 words and the longest 74 words.

We then calculated the distance of each sentence to all other sentences with 7 different distance metrics: Chebyshev, Cosine, Euclid, Hamming, City Block, Correlation and Spearman. For each sentence, we selected two sentences with the least distance calculated by each metric to be marked by the human annotators via a user interface as shown in Fig.1 with five marking options as: Paraphrase, Entailing, Entailed, opposite, not-Paraphrase. The target statement is shown on top of the screen. Three annotators labelled each sentence in the list with a label provided via pull down menu. Annotators marked the sentence as paraphrase when they believe the sentence gives the same or very similar meaning with the target sentence. Annotators marked the sentence as entailed when they think that it can be inferred from the target statement and entailing vice versa. The final decision is made if at least two annotators marked the same choice.

4 Results

Each of 7 similarity metric considered proposes 2 candidates as paraphrase for every sentence in the set of 399 sentences. So, we have $399 * 14 = 5586$ sentences proposed as candidates. By further eliminating the same candidates proposed by different metrics we are left with 2472 sentences to be annotated by humans. Symmetric measures of the Kappa analysis for three annotators with initial “S”, “T”, and “U” are given in Table 2. The second row shows the symmetric agreement ratios between the annotators when the annotations are quantized to 1 (Paraphrase - Paraphrase /entailed /entailing) and 0 (Not Paraphrase - opposite /notParaphrase). The third row shows the symmetric agreement ratios between the annotators on 5 scale detail.

The sentences that are symmetrically labelled as paraphrases of each other, i.e., for given sentences A and B, if A is labelled as paraphrase of B and B is labelled as paraphrase of A, are interpreted as true paraphrases. The annotators have marked 147 (6%) pairs as paraphrases, 85 of which are bidirectional, and 62 of which are unidirectional.

Table 3 shows the percentage ratios (number of TP/(399*2)) of true positives of the metrics.

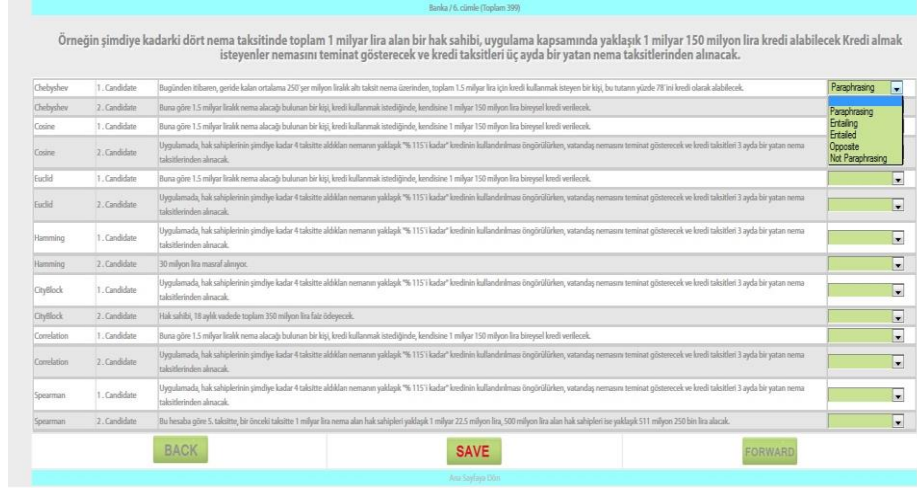


Fig. 1. Annotation interface

Table 2. Kappa results for Inter-annotator agreement

Ranking	S-T	S-U	U-T
Binary Scale: Paraphrase / notParaphrase	0.87	0.75	0.81
5 Scale: Paraphrase / entailed / entailing / opposite / notParaphrase	0.83	0.70	0.77

Table 3. True Positive Percentages of the Text Similarity Metrics

Sim-Metric	% of TP
Chebyshev	0,085
Cosine	0,343
Euclid	0,243
Hamming	0,243
CityBlock	0,248
Correlation	0,338
Spearman	0,333

As seen from Table 3 similarity metrics group into three. Chebyshev performs the least, in the middle performing group we have Euclid and Hamming and in the most performing group we have, Cosine, Correlation and Spearman.

Rus (2014) argues that paraphrase sentences extracted from the same news have large word overlap. Table 4 gives the word overlap ratios of the sentence pairs marked as paraphrase and not paraphrase by the annotators.

Table 4. The word overlap ratios of the sentence pairs

	Max	Min	Avg.
Paraphrase	0.98	0.0	0.49
Not paraphrase	0.86	0.0	0.12
All sentences	0.98	0.0	0.14

5 Conclusion

In this paper, we present preliminary work for the first stage, Turkish paraphrase corpus development, of a study with the ultimate goal of paraphrase recognition. The method is based on applying different text similarity metrics on the sentences driven from similar topics and choosing the sentences with the highest similarity scores as the candidates. Here we noticed that some of the methods are not functioning well for measuring similarity at sentence level for Turkish. We will continue processing BilCon2005 Corpus, topic by topic, with improved data cleaning, pre-processing and choice of text similarity metrics.

As it is argued in [10], paraphrase sentences extracted from the same news have large word overlap which is in conflict with the definition of “expressing the same meaning with different (own) words”. Most of the words in our sentence pairs overlap as in MSRP sample sentences given below.

Text A: York had no problem with MTA’s insisting the decision to shift funds had been within its legal rights.

Text B: York had no problem with MTA’s saying the decision to shift funds was within its powers.

Our next goal is to foster this corpus with sentences obtained from several other sources with different nature. We are planning to include sentences from different translations of the same texts, paraphrases from Turkish Language level assessment exams in order to enrich the corpus by including broader range of linguistic phenomena [10] and challenge the problem on a wider space.

Acknowledgement

This work is carried under the grant of TÜBİTAK – The Scientific and Technological Research Council of Turkey to Project No: 114E126, Using Certainty Factor Approach and Creating Paraphrase Corpus for Measuring Similarity of Short Turkish Texts.

References

1. Quirk, C., Raghavverda, U., Arul, M.: Generative models of noisy translations with applications to parallel fragment extraction. In: MT Summit XI, Copenhagen, Denmark, pp. 321–327 (2007)
2. Regneri, M., Koller, A., Pinkal, M.: Learning Script Knowledge with Web Experiments. In: the Association for Computational Linguistics (2010)
3. Wang, R., Callison-Burch, C.: Paraphrase fragment extraction from monolingual comparable corpora. In: the ACL BUCC-2011 Workshop (2011)
4. Barzilay, R., Lee, L.: Learning to paraphrase: An unsupervised approach using multiple sequence alignment. In: HLT-NAACL 2003. (2003)

5. Dolan, B., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In: COLING 2004 (2004)
6. Quirk, C., Brockett, C., Dolan, W.: Monolingual Machine Translation for Paraphrase Generation. In: the 2004 Conference on Empirical Methods in Natural Language Processing, pp.142-149 (2004)
7. McCarthy, P., McNamara, D.: The user-language paraphrase challenge. In: Special ANLP topic of the 22nd International Florida Artificial Intelligence Research Society Conference, Florida (2008)
8. Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In Benno Stein, P., ed.: SEPLN 09 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09), pp.1-9 (2009)
9. Lintean, M., Vasile, R., Azevedo, R.: Automatic Detection of Student Mental Models during Prior Knowledge Activation in MetaTutor. *International Journal of Artificial Intelligence in Education* 21(3), 169-190. (2011)
10. Rus, V., Banjade, R., Lintean, M.: On Paraphrase Identification Corpora. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, pp.2422-2429 (2014)
11. Agirre, E., Cer, D., Diab, M., González-Agirre, A., Guo, W.: SEM 2013 shared task: Semantic Textual Similarity. In: the Second Joint Conference on Lexical and Computational Semantics, Atlanta, Georgia, USA, vol. volume 1. Association for Computational Linguistics, pp.32-43 (2013)
12. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In Quiñero-Candela, J. ., ed.: *Machine Learning Challenges. Lecture Notes in Computer Science*, vol. 3944, pp.177-190 (2006)
13. Rus, V., McCarthy, P., Lintean, M., McNamara, D., Graesser, A.: Paraphrase identification with lexico-syntactic graph subsumption. In Sutcliffe, D., ed.: the 21st International Florida Artificial Intelligence Research Society Conference, pp.201-206 (2008)
14. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research* 38, 135-187 (2010)
15. Brockett, C., Dolan, W.: Support Vector Machines for Paraphrase Identification and Corpus Construction. In: Third International Workshop on Paraphrasing (IWP2005) (2005)
16. Barzilay, R., Ely, M.: Extracting paraphrases from a parallel corpus. In: the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France, pp.50-57 (2001)
17. Bannard, C., Callison-Burch, C.: Paraphrasing with Bilingual Parallel Corpora. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, Michigan, pp.597-604 (2005)
18. Dolan, W., Brockett, C.: Automatically Constructing a Corpus of Sentential Paraphrases. In: Third International Workshop on Paraphrasing (IWP2005)

19. Chen, D., Dolan, W.: Collecting Highly Parallel Data for Paraphrase Evaluation. In: the proceedings of The 49th Annual Meetings of the Association for Computational Linguistics (ACL), Portland (2011)
20. Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., Pinkal, M.: Grounding action descriptions in videos. Transactions of the Association for Computational Linguistics 1, 25-36 (2013)
21. Regneri, M., Wang, R.: Using discourse information for paraphrase extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.916-927 (2012)
22. Bernhard, D., Gurevych, I.: Answering learners' questions by retrieving question paraphrases from social Q&A sites. In: the Third Workshop on Innovative Use of NLP for Building Educational Applications, Ohio, USA, pp.44-52 (2008)
23. Can, F., Kocberber, S., Baglioglu, O., Kardas, S., Ocalan, H., Uyar, E.: New event detection and topic tracking in Turkish. Journal of the American Society for Information Science and Technology 61(4), 802-819 (2010)

An Approach to Content Extraction from Scientific Articles using Case-Based Reasoning

Rajendra Prasath and Pinar Ozturk

Department of Computer and Information Science
Norwegian University of Science and Technology
Sem Slands vei 9, 7491 Trondheim, Norway
drrprasath@gmail.com; pinar@idi.ntnu.no

Abstract. In this paper, we present an efficient approach for content extraction of scientific papers from web pages. The approach uses an artificial intelligence method, Case-Based Reasoning(CBR), that relies on the idea that similar problems have similar solutions and hence reuses past experiences to solve new problems or tasks. The key task of content extraction is the classification of HTML tag sequences where the sequences representing navigation links, advertisements and, other non-informative content are not of interest when the goal is to extract scientific contributions. Our method learns from each experience with the tag sequence classification episode and stores these in the case base. When a new tag sequence needs to be classified, the system checks its case base to see whether a similar tag was experienced before in order to reuse it for content extraction. If the tag sequence is completely new, then it uses the proposed algorithm that relies on two assumptions related to the distribution of various tag sequences occurring in the page, and the similarity of the tag sequences with respect to their structure in terms of levels of the tags. Experimental results show that the proposed approach efficiently extracts content information from scientific articles.

Keywords: Literature Based Knowledge Discovery, Information Extraction, Similar Pattern Mining, Case Based Reasoning

1 Introduction

Humans manage not to get distracted by advertisements, navigational menus, recommended articles, etc when they read a web page because they quickly learn where each of these noisy portions of a page is usually located and how does each portion look like. The ability of distinguishing the actual content from the rest of the page is necessary for computers as well because the noisy text may adversely affect the search and text mining results. We are specifically interested in text mining of scientific papers for the purpose of knowledge discovery. Hence, we would like the computer to process only the content of the scientific articles crawled from scientific publishers. This would have been an easy task if all journal publishers had used the same structure where each type of noisy element was

always located at the same position on a page. However, the fact is that each publisher do it differently and the same publisher may use a different layout for each journal they are publishing. In addition, each publisher often changes the layout they are using over time. Hence, at any moment there are a large number of layouts for the scientific papers on the Web. This makes it necessary for the computer to learn from its experiences and quickly recognize a page layout if it was deciphered earlier.

Several web page information extraction techniques have been introduced to automatically extract the main content using various hybrid approaches that apply segmentation methods based on heuristics or visual features to identify the main content of the webpage. Finding the features that are more salient for recognizing the main content is a challenging task. This problem is much complicated when we attempt to extract coherent content from scientific research articles. In this paper, we present an approach for learning to extract the main content using case based reasoning, an artificial intelligence method for problem solving, that applies incremental learning and reuses its knowledge about specific patterns that it has observed earlier for the efficient extraction of the main content from scientific articles.

This paper is organized as follows: After presenting a brief review of the related work in Section 2, we describe the objectives of the proposed work in Section 3. Then in Section 4, we briefly present the artificial intelligence method called Case-Based Reasoning. In Section 5, the proposed approach is organised in three subtasks: *corpus acquisition*, *patterns extraction*, and *content extraction*. Section 6 describes our experimental results. Finally Section 7 concludes the paper with a brief discussion on future work.

2 Related Work

Web content extraction is very well investigated in the literature [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Many of these approaches apply techniques based on certain heuristics, machine learning or site specific solutions like rule based content extraction, DOM tree parsing, Text graph or Link Graph or vision based models, or NLP features like N -grams or shallow text features like number of tokens, average sentence length and so on.

Cai *et al.* [1] presented a compilation of various approaches to handle web content and approaches to perform web mining related tasks. Debnath *et al.* [2] presented an approach for the automatic identification of informative sections of webpages. This approach segments the given web document into several constituent web page blocks and then applies several algorithms to identify the text blocks of the primary content section and filters out the non-informative content.

Finding the content portion of a web page is not straight forward across multiple websites. Many solutions to this problem involve customized rules or scripts to extract the content. Writing separate extraction routines for each website is also time-consuming and many times fail when the web page

layout changes over time. Gibson *et al.* [3] presented an approach that models the content identification problem as a sequence labelling problem using a Conditional Random Field sequence labelling model.

Chakrabarti *et al.* [4] presented an approach for segmenting a webpage into visually and semantically cohesive parts based on weighted graph technique. The weights between two nodes in the Document Object Model (DOM) tree should be placed together or apart in the segmentation. This framework learns the weights from manually labelled data and helps in the segmentation process.

Kohlschütter *et al.* [6] proposed a boilerplate detection algorithm using shallow text features. In this work, non-informative text portions called *boilerplates* have been detected by using a small set of shallow text features. These shallow text features help in classifying the individual text elements in a webpage. Boilerplate creation process is guided by a stochastic model that uses features like: *average word length*, *average sentence length*, *text density*, *link density*, *number of words* and combination of local features.

Sleiman and Corchuelo [9] presented a comprehensive survey of content extractors from web documents. They also proposed an unsupervised web data extractor using Trinary trees. In this approach, two or more web documents from same source are taken to learn a regular expression which is then used to extract data from similar documents. This works on the hypothesis that the input source documents share some common patterns. In another work[13], the same authors proposed an unsupervised information extractor that finds and removes shared token sequences amongst these web documents until finding the informative content.

Yao and Zuo [14] presented a classification based approach to perform webpage content extraction. In this work, a set of relevant features is selected for each text block in the HTML document and then using a Support Vector Machine (SVM) classifier, each text block is classified as either content block or non-content block. Most recently, Wu *et al.* [12] formulated the content identification problem as a DOM tree node selection problem. Using multiple features from DOM node properties, a machine learning model is trained and a set of candidate nodes is selected based on the learning model. The authors observed that the actual content is found in a spatially continuous block. They developed a grouping approach to filter out noisy content and pick the missing data for the candidate nodes.

3 Objectives

With the rapid growth of various research activities and scientific document publishers, the volume of scientific articles keep increasing very fast. Scientist need text mining and search systems to support their knowledge discovery endeavour. The users often search for a topic, a particular researcher/author, combination of topic and year, etc. while in text mining specific entities (e.g., temperature, pH, phytoplankton) or topics (e.g., impacts of increasing temperature on phytoplankton growth) may be focussed on. The input to a text

mining system is the content of the scientific papers on the Web. Hence we aim to develop a content extraction approach that is effective and domain independent.

Figure 1 illustrates a web page of a scientific paper consisting of various blocks where same type of blocks are marked with the same number. Among these blocks, some are just noisy blocks such as navigational links (marked with number 1), banners, personalized menus and advertisements. We need to identify the blocks of interest that cover various parts of a scientific article (marked with number 7 in Figure 1) such as author details, affiliation details, abstract, keywords, headers, sections, subsections, figures, tables, references, contact details, acknowledgements, and so on. To distinguish these blocks from the noisy blocks, we refer the former as *valid* or *informative* blocks of the page. Our goal is to differentiate between blocks with informative content from the ones with non-informative content, and only extract the content of the informative parts of a web page. Figure 1 illustrates how the non-informative or non-interesting parts of the page are spread over the page. A Web page is represented in HTML format which can be translated into a representation consisting of a set of HTML tag sequences where only some correspond to the valid content. The valid content identification can be casted as recognition of valid tag sequences. Tag sequence recognition, as seen in Section 2 is a challenging and costly process based on some regularities and heuristics pertinent to web page layouts.

Our objective is to develop a content extraction system that can efficiently identify valid tag sequences in the web page through acquiring a new experience each time it encounters a tag sequence the first time, and reusing this knowledge when a new tag sequence to be classified (as informative or non-informative) was seen earlier.

4 Case-Based Reasoning

Before describing the proposed approach to content extraction from scientific articles, we give a brief description of the case-based reasoning method which is the backbone of our approach.

Case-Based Reasoning is problem solving paradigm that solves new problems based on the solutions of similar past problems [15, 16, 17]. The past cases are stored in a case base where each *case* is a contextualized piece of experience and has a problem description part which can be represented either in vector representations, structured representation or in text representations[17], and a solution part. CBR consists of four important steps: *Retrieve*, *Revise*, *Reuse* and *Retain* (see Figure 2). When used for a classification task, the problem description will consist of the features describing the problem while the solution will be the class that it belongs to while in a planning task, the solution will be a sequence of actions to take. The principle idea for all task is that similar problems will have similar solutions. Hence, if a new problem description matches with an already solved problem existing in the case base it is retrieved to be re-used for solving the new problem. CBR allows incremental learning; when the current



Fig. 1. Scientific article crawled from Nature Publishers. Blocks marked with X are not part of the main article content and need to be filtered out. Only those marked with number 7 are "valid/informative" for our purpose.

problem to be solved does not already exist in the case base, it is retained in the Case Base after it is solved in order to be reused to solve future similar problems.

We use CBR for gathering informative tag sequence experiences and using these in order to increase the efficiency of the system in classifying tag sequences in a new web page.

5 Proposed Approach

The overall approach consists of three main components, as shown in Figure 3. The first is the *corpus acquisition* component which converts the HTML

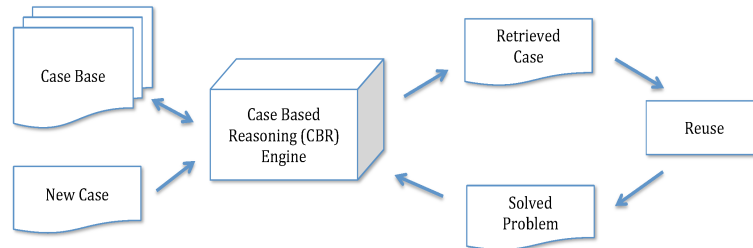


Fig. 2. Case Based Reasoning Paradigm

representation of a web page that includes a scientific text into a representation consisting of a set of HTML-tag sequences where each tag sequence corresponds to a block (labelled with a number in Figure 1) in the page. Figure 1 illustrates different blocks of a web page of a scientific paper. The next component, *pattern extraction*, classifies a tag sequence as either informative/interesting or non-informative. The last component is *content extraction* which extracts the content of the blocks in the web page represented by valid patterns. The key issue is to identify tag-sequences in HTML representation that are corresponding to the valid content, e.g., sections of the scientific article. For the knowledge discovery task, we may also be interested in some information about the article, such as authors and the journal it is published in (marked with uppermost 7 in Figure 1).

Our approach includes an algorithm that classifies a tag sequence as representing valid or noisy data, through an elaborative comparison of all tag-sequences in the concerned webpage (explained in detail in Section 5.2.2. An important attribute of the proposed approach is that it can take advantage of its earlier experiences about tag sequence classification. The tag sequence classification in Section 5.2.2 is used only when the new tag to be classified is not seen before, that is, a case similar to this one does not exist in the case base of the system.

The overall approach is represented as a pseudocode in Algorithm 1 while Figure 3 gives a more detailed picture of the approach. In the rest of Section 5, we describe three main components and their sub-components in detail.

5.1 Corpus Acquisition

5.1.1 Crawling We have identified the set of seed urls for the articles of the journals that are considered under three different categories in the specified domain by domain experts: *Level - 2: high impact journal*, *Level - 1: low important journal* and *Level - 0: other journals*. The choice of the categories may vary across researchers in the same domain and the impact of the journal is also selected based on their scientific impact factor and the citation of the articles published in that journal. So it is a co-related factor associated with the choice of the publication channel of the domain experts in the specific domain. We have used the collected seed urls and performed crawling of scientific articles. In

Algorithm 1 The proposed approach

Input: A collection of scientific articles, each in HTML format

Description:

```
1: Preprocess the input HTML document into a well structured HTML document
2: Parse the HTML document into a set of tag-sequences
3: for each tag sequence of the document do
4:   Match the tag sequence with the cases in the case base
5:   if a similar case (i.e, a tag sequence / pattern) is retrieved then
6:     Perform content extraction with the retrieved pattern
7:   else
8:     Identify the candidate tag sequence / pattern using count- and
       level-assumptions and retain in the case base
9:     Extract the content using the identified pattern
10:  end if
11: end for
```

Output:

- a) A case base of tag patterns;
 - b) Extracted blocks of the main content of scientific articles.
-

this experiment, we have considered only a few publishers popular in publishing journals of high impact factors. After crawling the scientific articles, we have stored the original content of the articles in HTML format. We further use these articles to perform the extraction of the main content for exploratory purposes.

5.1.2 Preprocessing The scientific articles are crawled in HTML format from various publishers. An HTML document in the crawled data may have noisy contents like advertisements, banners, recommended articles, navigational links, copyright information and so on. We perform the metadata extraction of the article in the form of an attribute-value pairs. Additionally, we use the pattern extractor that explores the structure of the main article content, filters out the patterns with class references of similar blocks of the main text content of the scientific article. The text content of such similar patterns can represent coherent blocks (or sections) of a scientific article. We have adapted the preprocessing steps similar to the one given in [18].

In Section. 5.3.1, we explain the detailed procedure to segment an article (in HTML format) into coherent blocks.

5.1.3 Parsing We apply top down parsing by traversing over the HTML tags (nodes) in the well structured HTML document. At each node, we populate the tag sequence of the underlying subtree and we repeat this process until we process all open tags till the end of the web document. Each of these tag sequences represent a specific part of the layout. So the output of the parsing is a set of tag sequences.

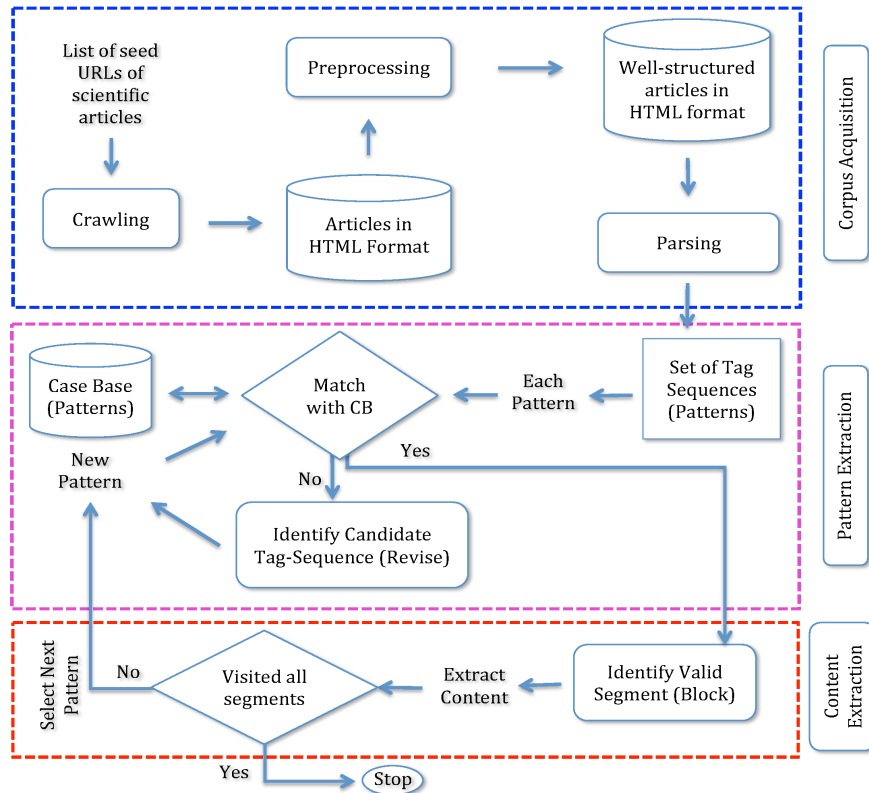


Fig. 3. The proposed CBR-based approach to content extraction.

5.2 Pattern Extraction

5.2.1 CBR Retrieval When a new tag sequence in the document is to be classified as a valid or invalid pattern, the system first tries to retrieve a case with a similar pattern from the case base. Since we preserve the order of tags in the sequence, the similarity measure applied here is based on substring matching and gives higher preference to the matches that occur at higher levels of tags. If the system finds a similar case, then the current tag sequence is valid indicating that the corresponding content should be extracted. If there is no similar case in the case base, then the system resorts to the classification algorithm explained in Section 5.2.2. If the tag sequence is classified as valid by this algorithm, then it is learnt and retained in the case base.

5.2.2 Identifying Candidate Tag Sequences Our approach is based on two important assumptions that we call as *count* assumption and *level* assumption. The *count assumption* expresses that sections of a scientific article are assumed to follow a similar tag sequence. This means that this type of tag

sequence occurs as many times as the number of sections in the article. This in turn, means that the tag sequence with the most occurrence on a web page of the scientific article is highly probable to belong to the scientific text. The *level assumption* suggests that for an article with several sections, even when the whole tag sequence of each section does not fully matches with that of other sections, a large portion of tag sequences of all sections may be overlapping at different levels that correspond to different subsections. Tag sequences of two sections that overlap in higher levels are considered to be more similar. The more such higher levels they overlap, the more similar they would be. Putting these two assumptions together, tag sequences that are similar according to the level assumption that occurs most often in a web page are good candidates to represent a section of a scientific text. These candidate patterns are used to populate the case base which was empty at the initial state.

5.3 Content Extraction

5.3.1 Segmenting Scientific Article into Blocks

For each valid tag sequence in the document, we scan through the document and identify all matching blocks. Each block is segmented according to the tag sequence.

5.3.2 Extracting Text from Blocks Step 1: We extract the meta tags by matching META attributes and values of the HTML document. We collect most of the available metadata of articles from the HEAD tag.

Step 2: In this step, we focus on the content part of the scientific article. We match the tag sequences to the corresponding blocks of the scientific articles. We extract the text from matching blocks.

6 Experimental Results

6.1 Corpus

We have crawled scientific articles published by various publishers and the details are listed in Table 1. In this experiment, we have crawled both open access articles and articles with restricted access option (subscription based). The articles collected from journals: *Climate Change*, *Ecosystems* and *The ISME Journal* are of open access and the articles collected from journals: *Ecology Letters*, *Nature Geoscience* and *Nature Communications* are of restricted access.

Scientific research articles in the areas of Marine Science, Climate Science and Environmental Science are the outcome of research experiments and observations describing the underlying complex theories and models. Lots of variables like chemical compositions, metals, living organisms and other species are involved in such environmental models. To explore the facts behind these variables and find out how the variables are quantified in the presence of an associated variable or an event in the marine food chain. These journals are suggested by the

Publisher	Journal Name	# Articles
Springer	Climate Change	473
Springer	Ecosystems	73
Macmillan Publishers	The ISME Journal	113
Wiley	Ecology Letters	1650
Macmillan Publishers	Nature Geoscience	225
Macmillan Publishers	Nature Communications	2,741

Table 1. Data used in our experiments

domain experts. This data is the part of the ongoing research work on the literature based knowledge discovery tasks and is a part of the corpus used in the OCEAN-CERTAIN¹ project.

6.2 Evaluation Measures

In this section, we present the evaluation measure of the proposed approach. We used the following measures to quantify the goodness of the extracted blocks: *Purity* and *Accuracy*. Let us define these measures now.

Purity is used to quantify the number of noise free blocks that are extracted from the given input document. We define *Purity* as the ratio between the number of blocks without noisy content and the total number of blocks extracted.

Accuracy is used to quantify the number of content blocks that are correctly extracted from the given input document. We define *accuracy* as the ratio between the number of blocks correctly extracted and the total number of blocks in the input document.

6.3 Discussion

We have presented our preliminary results in this work. We have considered top 20 articles in each journal and manually evaluated the efficiency of the proposed information extraction approach. The results are tabulated in Table. 2.

Journal Name	Purity	# Accuracy
Climate Change	91.2 %	95.43 %
Ecosystems	93.57 %	88.52 %
The ISME Journal	90.72 %	91.83 %
Ecology Letters	85.27 %	89.39 %
Nature Geoscience	94.26 %	93.41 %
Nature Communications	96.37 %	95.43 %

Table 2. Purity and Accuracy scores of the selected journals (averaged over 20 documents)

¹ OCEAN CERTAIN - A project funded by European Union and lead by NTNU with 11 partners from 8 European countries and Chile and Australia. http://cordis.europa.eu/project/rcn/110540_en.html

Since Nature articles are well organized into coherent sections, each section heading and subheadings are explicitly marked up. It also helps to filter out figure captions along with the table related data from the main content. In journals published by other publishers, the wrapper has to identify and extract this information with more efforts. Additionally, we observed majority of the articles consists of citation and copyright blocks along with the main content. It also reduces the accuracy of the proposed approach.

7 Conclusion

In this work, we have proposed an approach that performs learning to extract vital information from scientific articles based on a Case-Based Reasoning approach. In a web page of a scientific paper, there will be more number of article related blocks than other, noisy parts. Since tag sequences representing the segments of the scientific text will be similar, the tag sequence with highest number of occurrence in the page is a good candidate to represent the content of our interest. Here we attempted to use this heuristic without relying on any linguistic or semantic expertise to extract the right content from different publishers. Experimental results carried out on the subset of the scientific articles collection show that the proposed approach effectively extracts the relevant information that is useful to do knowledge discovery. Further we need to recognize patterns that belong to the specific parts of a scientific article, that is, whether the pattern belongs to author(s) name, title or the main body of the scientific content. To achieve this, we plan to extend the CBR part of the system and add the label of the tag sequence that specifies the part of the scientific article.

Acknowledgment

Authors gratefully acknowledge the support of European Commission through OCEAN-CERTAIN under the grant no. 603773.

References

- [1] Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Extracting content structure for web pages based on visual representation. In: Proceedings of the 5th Asia-Pacific Web Conference on Web Technologies and Applications. APWeb'03, Berlin, Heidelberg, Springer-Verlag (2003) 406–417
- [2] Debnath, S., Mitra, P., Pal, N., Giles, C.L.: Automatic identification of informative sections of web pages. *IEEE Trans. on Knowl. and Data Eng.* **17**(9) (September 2005) 1233–1246
- [3] Gibson, J., Wellner, B., Lubar, S.: Adaptive web-page content identification. In: Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management. WIDM '07, New York, NY, USA, ACM (2007) 105–112

- [4] Chakrabarti, D., Kumar, R., Punera, K.: A graph-theoretic approach to webpage segmentation. In: Proceedings of the 17th International Conference on World Wide Web. WWW '08, New York, NY, USA, ACM (2008) 377–386
- [5] Pasternack, J., Roth, D.: Extracting article text from the web with maximum subsequence segmentation. In: Proceedings of the 18th International Conference on World Wide Web. WWW '09, New York, NY, USA, ACM (2009) 971–980
- [6] Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining. WSDM '10, New York, NY, USA, ACM (2010) 441–450
- [7] Sleiman, H.A., Corchuelo, R.: An unsupervised technique to extract information from semi-structured web pages. In Wang, X.S., Cruz, I.F., Delis, A., Huang, G., eds.: Web Information Systems Engineering - WISE 2012 - 13th International Conference, Paphos, Cyprus, November 28-30, 2012. Proceedings. Volume 7651 of Lecture Notes in Computer Science., Springer (2012) 631–637
- [8] Sleiman, H.A., Hernández, I.: A framework for populating ontological models from semi-structured web documents. In Atzeni, P., Cheung, D.W., Ram, S., eds.: Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings. Volume 7532 of Lecture Notes in Computer Science., Springer (2012) 578–583
- [9] Sleiman, H.A., Corchuelo, R.: A survey on region extractors from web documents. *IEEE Trans. Knowl. Data Eng.* **25**(9) (2013) 1960–1981
- [10] Sleiman, H.A., Corchuelo, R.: Trinity: On using trinary trees for unsupervised web data extraction. *IEEE Trans. Knowl. Data Eng.* **26**(6) (2014) 1544–1556
- [11] Crescenzi, V., Merialdo, P., Qiu, D.: Crowdsourcing large scale wrapper inference. *Distributed and Parallel Databases* **33**(1) (2015) 95–122
- [12] Wu, S., Liu, J., Fan, J.: Automatic web content extraction by combination of learning and grouping. In: Proceedings of the 24th International Conference on World Wide Web. WWW '15, New York, NY, USA, ACM (2015) 1264–1274
- [13] Sleiman, H.A., Corchuelo, R.: TEX: an efficient and effective unsupervised web information extractor. *Knowl.-Based Syst.* **39** (2013) 109–123
- [14] Yao, J., Zuo, X.: A machine learning approach to webpage content extraction. <http://cs229.stanford.edu/proj2013/YaoZuo-AMachineLearningApproachToWebpageContentExtraction.pdf> (2013) Online; Last accessed 23 February 2016.
- [15] Kolodner, J.: Case-based Reasoning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
- [16] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* **7**(1) (March 1994) 39–59
- [17] Bergmann, R., Kolodner, J., Plaza, E.: Representation in case-based reasoning. *Knowl. Eng. Rev.* **20**(3) (September 2005) 209–213
- [18] Prasath, R.R., Öztürk, P.: Similarity assessment through blocking and affordance assignment in textual cbr. In: Proc. of the Reasoning from Experiences on the Web. WebCBR 2010, Alessandria, Italy, Web CBR (2010) 51–160

Event Causality Extraction from Natural Science Literature

Biswanath Barik, Erwin Marsi, Pinar Öztürk

Department of Computer and Information Science
Norwegian University of Science and Technology
{biswanath.barik, emarsi, pinar}@idi.ntnu.no

Abstract. We aim to develop a text mining framework capable of identifying and extracting *causal dependencies* among *changing variables* (or *events*) from scientific publications in the cross-disciplinary field of oceanographic climate science. The extracted information can be used to infer new knowledge or to find out unknown hypotheses through reasoning, which forms the basis of a knowledge discovery support system. Automatic extraction of causal knowledge from text content is a challenging task. Generally, the approaches of causal relation identification proposed in the literature target specific domain such as online news or biomedicine as the domain has significant influence on causality expressions found in the domain texts. Therefore, the existing models of causality extraction may not be directly portable to other/new domains. In this paper, we describe the nature of causation observed in climate science domain, review the state-of-the-art approaches in causal knowledge extraction from text and carefully select the methods and resources most likely to be applicable to the considered domain.

Keywords: causal relation, information extraction, relation extraction, knowledge discovery

1 Introduction

Climate change is a major concern in recent years and has various direct and indirect effects in day-to-day life. Global warming, intensified with various anthropogenic activities, have serious impacts on the precious climate system of this blue planet. As a result, the pattern of the climate has been changing rapidly. The other associated natural processes and systems are, in consequence with climate change, being affected significantly. Changes in the ecological system of the marine environment is one such system.

Various studies in the domain of marine science, climate science, environmental science and other related fields of Earth Science report significant *changes* in recent years in several *parameters* (i.e., quantitative variables) of the ocean environment. *Sea-surface temperature, bottom-water temperature, direction of ocean circulation, acidity, pH, alkalinity* and *CO₂ concentration level in water* are a few examples of variables observed to change significantly in past years.

Significant changes of such variables have impact on *phytoplankton growth rate* which indirectly affects marine *food web* - a complex feeding network of different species (i.e., who eats whom) living in the marine environment. Along with this, the efficiency of the *biological pump*, the ocean's ability to absorb and store CO_2 as the food web transfer parts of the biomass deeper into the ocean, is degraded.

The research publications in climate science, marine science and environmental science are the authentic sources of information describing various theories and models consisting of changing variables (or *events*) and their complex interactions. The elementary interactions among the events may be in the form of correlations, causal relations or the positive/negative feedback cycles consisting of sequence of events. Identifying and extracting valuable interactions from the scientific articles and combining them to explore various hidden connections among the events can help to better understand the functionality of various processes of the domain and their dependencies. However, the sheer volume of the articles limits scientists and policy makers to collect useful information by reading the articles in due time. Human cognition, on the other hand, may be another limit for recognizing and interpreting various cross-domain knowledge fragments. Therefore, an automated knowledge discovery support system is needed to quickly process the vast collection of research articles, extract useful knowledge fragments and produce new insights, hypothesis or discover unknown knowledge by combining the extracted knowledge units.

To distill essential factual knowledge from unstructured text content of research papers, the text mining techniques are successfully applied in the domain of bio-medicine. Significant advancements have been observed in identifying named entities [9], detecting events [28], coreference resolution [1] and causal relations extraction [26]. With the use of domain dependent Natural Language Processing(NLP) tools like Part-Of-Speech (POS) tagger, shallow and full sentence parsers for syntactic analysis of the text content, biomedical text mining is capable of providing a platform where researchers can query on the vast database of research articles of the domain. The researchers don't need to bother how many papers the system needs to process to find or infer the required answer, or how did the system do it?

In the domain of Natural Science, specifically in the cross-disciplinary domain of oceanographic climate science, our goal is to develop a (literature-based) knowledge discovery support system to facilitate a large community of scientists and researchers. The major challenges we face are the lack of resources like task specific annotated corpora, indexed literature databases covering the entire field, domain dependent NLP tools with good accuracies and knowledge resources (ontologies) as our target domain is almost unexplored. The resources and tools developed in biomedicine domain are not directly usable due to domain difference as shown in [24, 25]. Therefore, to meet this goal, a constant effort is being employed to develop resources and tools. In [25], authors describe an annotation scheme to annotate quantitative variables, their change events, correlations and causal relations among change events, and feedback loops from the abstracts and full-text journal papers collected from the nature publication. In [24], authors

automatically identify and extract variables and their direction of changes using a tree pattern matching technique and generalise these variables by progressive pruning of syntax tree using tree transformation operations.

In this research direction, our target is to develop a *causality extraction model* in oceanographic climate science domain where the causal relation among change events (as described in Section 2) can be automatically extracted from the scientific publications. The causal relations extracted from a collection of research papers of the domain can then be used for causal reasoning with the help of domain knowledge to discover new facts or unknown hypotheses of the domain. Such a reasoning system can provide a better information search (semantic search) capability to the scientists and researchers to efficiently access vast database of publications. In this paper, we explore the existing methods and algorithms of event causality identification and carefully select the methods and resources most likely to be applicable to oceanographic climate science domain with a proposed work plan.

The paper is organized in the following way. Section 2 describes the causality in oceanographic climate science domain. Section 3 shows the nature of causation observed in the considered domain. Section 4 describes the existing approaches for handling causality in text content. Section 5 discusses the suitability of the existing algorithms in this domain and the proposed work plan. Section 6 concludes the paper.

2 Problem Description

Causality extraction from text content is a fundamental task towards the desire of developing literature-based knowledge discovery support system. In climate science domain, an *event* is defined as: “a change is an event in which the value of a quantitative variable is changing” [25]. Causal relation, in general, is a semantic relation between two events where the occurrence of one event (called *cause event*) causes the occurrence of the other event (referred as the *effect event*). Figure 1 shows a typical example of causal relation between two events E_1 and E_2 , where E_1 is “Reduced calcification of marine plankton” and E_2 is “increased atmospheric CO_2 .”

The causality between these two events is expressed explicitly by the causality marker (or cue phrase) “in response to”. All the examples (Fig. 1 - Fig. 4) of causal relation in climate science domain are taken from the pilot annotation described in [25].

We have mentioned earlier that climate science is a new text mining domain. The necessary resources and domain specific NLP tools are not available in order to immediately and effortlessly build the pipeline for analyzing larger context and to just focus only on causal relation extraction module. Keeping in mind this limitation, we are interested to develop our causal relation extraction model incrementally. In the first step, we focus on identifying intra-sentence explicit causal relations. Some issues related to identifying causal relations within a single sentence where causation is explicitly expressed are described in the following

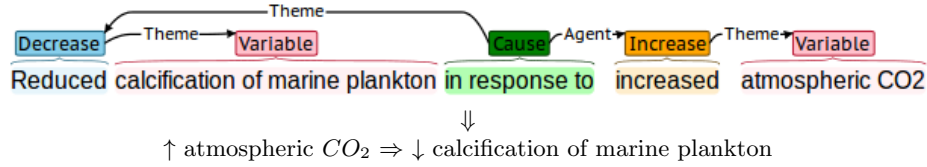


Fig. 1. Causality between two events in Natural Science literature

section. We have a plan to deal with inter-sentence causal relations by identifying discourse causal markers and resolving coreference issues, but this is not a focus in this paper.

3 Nature of Causation in Climate Science

We have studied various diversities and issues related to the notion of causation and how it may be expressed in natural language specifically in the research articles of climate science collected from Nature publications. These diversities need to be discussed in detail in order to develop a causal relation extraction model in this domain. The issues are described below:

1. **Multi-Event Participation in Causation:** Causal relation is, in general, a binary relation. It is a relation between two events: the cause event and the effect event as described in figure 1. However, it is observed that in causal implication, events can participate in the antecedent and/or in the consequence part as shown in figure 2 below.

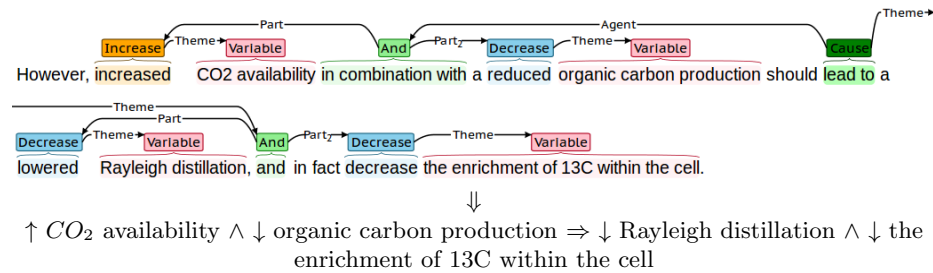


Fig. 2. Many-to-Many events causality

In this example, two events “increased CO_2 availability” and “reduced organic carbon production” are connected through the conjunction “in combination with” to form a larger (or composite) event and serve as a cause event to the causal relation signaled by “lead to”. Similarly, the events “lowered Rayleigh distillation” and “decrease the enrichment of ^{13}C within the cell” jointly forms the effect event. Therefore, causal relation in this example is a Many-to-Many(M:M) relation.

2. **Event Participating in Multiple-Causation:** Causal relations are sometimes expressed in a cascaded style where a single event can participate in more than one causal expressions in the same sentence. Figure 3 shows an example of such causation.

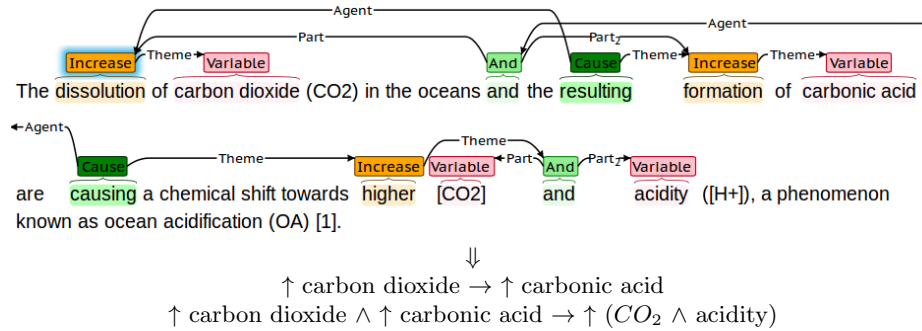


Fig. 3. Many-to-many event causality

In this example, the first causal relation expresses an 1:1 relation between events “↑ carbon dioxide” (cause event) and “↑ carbonic acid” (effect event). The second causal relation in the same sentence represented as a M:M event relation where the events of the first relation collectively constitute the cause event and the events “↑ CO₂” and “↑ acidity” are the effect events.

3. **Double Role of Causal Markers:** It is observed in other domains that the causal markers are often ambiguous i.e., they do not always express causality. Under certain context and semantic orientation, the markers express causality between events. This phenomenon is also true in climate science domain. However, we have noticed that the causal markers also *trigger* the change events along with its causation indication role in certain contexts.

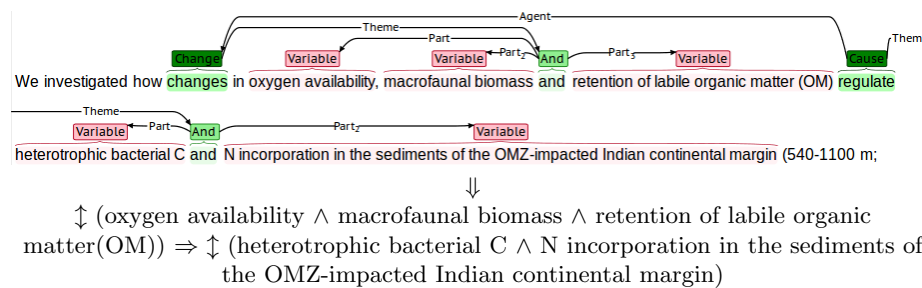


Fig. 4. Causality marker holds implicit trigger to event

In figure 4., the word *regulate* serves as a causal marker. Also, it serves as a *change* indicator (denoted as “↕”) to the variables “heterotrophic bacterial C” and “N incorporation in the sediments of the OMZ-impacted Indian continental margin”.

4 Causal Relation Extraction - A Brief Review

The general notion of causality is very broad. It has been studied in various fields of research like philosophy, statistics, cognitive science, linguistics, physics, economics, biology, medicine and so on. In computational linguistics, considerable amount of work has been done on automatic extraction of causal knowledge from text in general [5, 8, 22, 32] and in specialized domains like biomedical science [20, 26], or online news domain [11, 31]. However, causality knowledge extraction is a non-trivial problem till date. Many questions remain unsolved about the nature of causation. Also, causation is subjective - human judgement about causation is even conflicting in many instances (shows low inter-annotators agreement) and subject to the realization of the context [14]. In the following sub-sections we broadly categorize existing approaches of causality identification from text content found in literature.

4.1 Causality Extraction using Handcrafted Patterns

The initial attempts of causal relation extraction rely on knowledge-based inference techniques [18, 22, 35]. These works used linguistic patterns of causation along with manually crafted resources to detect causal relation hidden the context. In this research direction, Kaplan *et al.* [19] proposed a linguistic pattern-based approach for causal knowledge extraction where the resources like grammar, lexicon and domain-knowledge are hand-crafted for the target domain. Garcia [10] develop an approach where the causative verb patterns are extracted from *French* texts using handcrafted rules. In this experiment, the author found 25 causal relations and classified them with a precision of 85% using a semantic model based on “Force Dynamics” of Leonard Talmy [37]. Explicit causal relations are also identified from MEDLINE text database by Khoo *et al.* [20] using predefined linguistic patterns and achieved a precision about 68%. In this work, partially parsed verb linguistic patterns indicating causality relationships are matched on text to extract cause-effect information.

The causal relation extraction models based on linguistic patterns perform pretty well in restricted domains. However, rule creation is expensive and time consuming and it suffers from domain portability issue.

4.2 Semi-automatic Causal Pattern Learning

The other research direction explores semi-automatic learning of causal patterns from corpus with minimal (or no) domain knowledge. In this direction, Khoo *et al.* [21] developed an automatic system for extracting cause-effect relation

from newspaper texts using simple pattern-matching and without using linguistic clues and domain knowledge. In [12], Girju and Moldovan describe syntactic and semantic classification of cause-effect lexico-syntactic patterns found in English texts. They developed an approach to automatically identify lexico-syntactic patterns consisting of a pair of noun phrases connected by causative verbs ($\langle NP_1 \text{ verb } NP_2 \rangle$) that express the causal relations. Finally, a semi-automatic validation method is proposed to evaluate the extracted causal patterns. Marcu and Echihabi [23] classifies a sentence pair as ‘causal’ or ‘-causal’ by training a Naïve Bayes classifier on inter-sentence lexical pair probability. Girju [11] developed a decision tree based classifier on causality-annotated corpora, where the cue phrases are automatically extracted from WordNet [27] and also from the corpus, and achieved a precision of 73.91%. In [5], Blanco *et al.* first manually identify the syntactic patterns that may encode marked and explicit causation and found that the four most common relators encoding causation are *because*, *since*, *as* and *after*. Then they used decision tree based learning algorithm (an implementation of Bagging with C4.5 decision trees) to decide whether or not a pattern instance encodes a causation. However, this method is not able to detect the causes and the effects. Ittoo and Bouma [17] present a semi-supervised method for automatic extraction of high quality causal relations from domain-specific, sparse corpora. In this work, they initially acquire a set of explicit and implicit lexico-syntactic patterns from Wikipedia. Using some *seed* cause-effect patterns, the extracted patterns are then classified as causal or non-causal by measuring their reliability through computing point-wise mutual information between extracted patterns and seed patterns and ranking the extracted patterns accordingly. Finally, the extracted causal patterns are used to identify domain-specific causal relations.

4.3 Causality Prediction by Supervised Learning

In the supervised learning set-up, the domain corpus is needed to be annotated with events (or entities) and their causal relationships. Causality annotated corpora is then used to train supervised model for classifying a pair of events as causal or non-causal pair. In [11], the author manually annotates *Los Angeles Times corpus* based on explicit causal verbs (e.g., “to cause”). Using this annotated corpora and WordNet [27], the cue phrases are extracted, automatically. A decision tree classifier is then trained which detects the causality relation in news events with 73.9% precision and 88.7% recall. Beamer *et al.* [2] develop a support vector machine (SVM) based classifier trained on SemEval 2007 Task 4 corpus [13] and report an accuracy of 77.5% in identifying cause-effect noun pairs. SVM classifiers are also trained on annotated *Wall Street Journal (WSJ)* texts by [4,34]. In [29], the verb-pair rules are used to train Naïve Bayes (NB) and SVM classifier to identify causality from multiple Elementary Discourse Units (EDUs) and reported precision of 88% with NB and 89% with SVM.

4.4 Statistical Approaches

Existing corpus-based approaches to causality extraction use distributional characteristics of events, like co-occurrence features, object-sharing features, temporal features, distance features and so on. Machine learning based approaches are more robust than rule-based approaches and require less linguistic information and domain knowledge. Torisawa [38] developed a model for extracting commonsense inference rules from coordinate verb phrases based on co-occurrence and object sharing features. This work is further extended in [39] where the occurrence frequency of a single verb is emphasized and reported a relative improvement of 60 precision. Other approaches use predicate semantics [15, 16] and shared arguments [6, 7].

Since causation can be expressed in many different ways in natural language, the automatic recognition of causal relations is challenging. In recent works, researchers try to overcome this challenge by considering specific constructions like causation between verbs [2, 4, 8, 32], between verb-noun pairs [8, 33] and between two discourse segments [30, 36].

5 Proposed Work Plan

The approach of causal relation extraction using handcrafted rules/patterns is not suitable in our domain as our domain is cross-disciplinary in nature and creating causal patterns requires sufficient expertise of the domains. Also, in the rule-based causality identification approach, the rules created manually work well when the causality is obvious i.e., there has no ambiguity in identifying the causal indicators and the participating events. However, in Section 3, we have seen that causation in the considered domain is often complex in nature. Therefore, hand-made pattern-based causality detection is not feasible in this domain.

The supervised learning approach of causal relation is also not applicable as it requires large amount of causality annotated corpora of the domain which is a rather costly process. However, as we showed in Section 3 that an event can participate in more than one causal relation in some contexts, supervised classification approaches can be good candidates where the lexical, contextual, syntactic and semantic features can be well exploited for classifying a pair of events. We are under process of developing a small amount of causality annotated corpora for developing a baseline causal relation extraction model in our domain using supervised learning method like SVM. In the next step, we will experiment with various semi-supervised algorithms to improve the baseline performance using the annotated corpus and with large collection of unannotated texts.

The unsupervised way of measuring causal association between an event pair based on mutual information between them (PMI) and its variations like causal potential [3], Cause-Effect-Association (CEA) [8] reported good accuracy. We should explore the opportunities of such unsupervised approaches and evaluate their performance.

Our hypothesis pertinent to extraction of cascaded relations (issue 2 in Section 3) is that joint extraction of events and causal relations may better suit to cope with the complexity introduced by the inter-dependency between events and the causal expressions.

6 Conclusion

In this paper, we discuss about the necessity of the development causal relation extraction model in the cross-disciplinary field of climate science, marine science and environmental science. We describe the causality expressions found in oceanographic climate science domain and the issues need to be handled to develop a causal relation extraction model. We describe a brief survey of existing approaches of causal relation extraction from text data. Finally, we discuss about the suitability of existing causal relation extraction models in the considered domain and present the work plan.

References

1. Batista-Navarro, R., Ananiadou, S.: Adapting the cluster ranking supervised model to resolve coreferences in the drug literature. In: Proceedings of the Fourth International Symposium on Languages in Biology and Medicine (LBM 2011) (2011)
2. Beamer, B., Bhat, S., Chee, B., Fister, A., Rozovskaya, A., Girju, R.: Uiuic: A knowledge-rich approach to identifying semantic relations between nominals. In: Proceedings of the 4th International Workshop on Semantic Evaluations. pp. 386–389. Association for Computational Linguistics (2007)
3. Beamer, B., Girju, R.: Using a bigram event model to predict causal potential. In: Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing. pp. 430–441. CICLing '09, Springer-Verlag, Berlin, Heidelberg (2009)
4. Bethard, S., Corvey, W.J., Klingenstein, S., Martin, J.H.: Building a corpus of temporal-causal structure. In: LREC (2008)
5. Blanco, E., Castell, N., Moldovan, D.I.: Causal relation extraction. In: LREC (2008)
6. Chambers, N., Jurafsky, D.: Unsupervised learning of narrative schemas and their participants. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. pp. 602–610. Association for Computational Linguistics (2009)
7. Chambers, N., Jurafsky, D.: Unsupervised learning of narrative event chains. In: ACL. vol. 94305, pp. 789–797. Citeseer (2008)
8. Do, Q.X., Chan, Y.S., Roth, D.: Minimally supervised event causality identification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 294–303. Association for Computational Linguistics (2011)
9. Fukuda, K., Tamura, A., Tsunoda, T., Takagi, T.: Toward information extraction: identifying protein names from biological papers. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing pp. 707–718 (1998)

10. Garcia, D.: Coatis, an nlp system to locate expressions of actions connected by causality links. In: Proceedings of the 10th European Workshop on Knowledge Acquisition, Modeling and Management. pp. 347–352. EKAW '97, Springer-Verlag, London, UK, UK (1997)
11. Girju, R.: Automatic detection of causal relations for question answering. In: Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12. pp. 76–83. Association for Computational Linguistics (2003)
12. Girju, R., Moldovan, D.I., et al.: Text mining for causal relations. In: Proceedings of Florida Artificial Intelligence Research Society (FLAIRS). pp. 360–364 (2002)
13. Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., Yuret, D.: Semeval-2007 task 04: Classification of semantic relations between nominals. In: Proceedings of the 4th International Workshop on Semantic Evaluations. pp. 13–18. Association for Computational Linguistics (2007)
14. Grivaz, C.: Automatic extraction of causal knowledge from natural language texts. Ph.D. thesis, University of Geneva (2012)
15. Hashimoto, C., Torisawa, K., De Saeger, S., Oh, J.H., Kazama, J.: Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 619–630. EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
16. Hashimoto, C., Torisawa, K., Kloetzer, J., Sano, M., Varga, I., Oh, J.H., Kidawara, Y.: Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In: ACL (1). pp. 987–997 (2014)
17. Ittoo, A., Bouma, G.: Extracting explicit and implicit causal relations from sparse, domain-specific texts. In: International Conference on Application of Natural Language to Information Systems. pp. 52–63. Springer (2011)
18. Joskowicz, L., Ksiezzyk, T., Grishman, R.: Deep domain models for discourse analysis. In: Proc Annu AI Syst Gov Conf. pp. 195–200 (1989)
19. Kaplan, R.M., Berry-Rogghe, G.: Knowledge-based acquisition of causal relationships in text. *Knowledge Acquisition* 3(3), 317–337 (1991)
20. Khoo, C.S.G., Chan, S., Niu, Y.: Extracting causal knowledge from a medical database using graphical patterns. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. pp. 336–343. ACL '00, Association for Computational Linguistics, Stroudsburg, PA, USA (2000)
21. Khoo C., K.J., et al.: Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. In: *Literary & Linguistic Computing*. pp. 177–186 (1998)
22. Kontos, J., Sidiropoulou, M.: On the acquisition of causal knowledge from scientific texts with attribute grammars. *Internat. Journ. of Appl. Exp. Sys.* 4(1), 31–48 (1991)
23. Marcu, D., Echihiabi, A.: An unsupervised approach to recognizing discourse relations. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 368–375. Association for Computational Linguistics (2002)
24. Marsi, E., Oztürk, P.: Extraction and generalisation of variables from scientific publications. In: Proc. of EMNLP. pp. 505–511 (2015)

25. Marsi, E., Oztürk, P., Aamot, E., Sizov, G., Ardelan, M.V.: Towards text mining in climate science: Extraction of quantitative variables and their relations. In: Proceedings of the Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing, Reykjavik, Iceland (2014)
26. Mihăilă, C., Ananiadou, S.: Semi-supervised learning of causal relations in biomedical scientific discourse. *Biomedical engineering online* 13(2), 1 (2014)
27. Miller, G.A.: Wordnet: a lexical database for english. *Communications of the ACM* 38(11), 39–41 (1995)
28. Miwa, M., Thompson, P., Ananiadou, S.: Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics* 28(13), 1759–1765 (2012)
29. Pechsiri, C., Kawtrakul, A.: Mining causality from texts for question answering system. *IEICE - Trans. Inf. Syst.* E90-D(10), 1523–1533 (2007)
30. Pitler, E., Nenkova, A.: Using syntax to disambiguate explicit discourse connectives in text. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. pp. 13–16. Association for Computational Linguistics (2009)
31. Radinsky, K., Davidovich, S., Markovitch, S.: Learning causality for news events prediction. In: Proceedings of the 21st International Conference on World Wide Web. pp. 909–918. WWW '12, ACM, New York, NY, USA (2012)
32. Riaz, M., Girju, R.: Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In: Proceedings of the SIGDIAL 2013 Conference. p. 21–30. Association for Computational Linguistics, Association for Computational Linguistics, Metz, France (August 2013)
33. Riaz, M., Girju, R.: Recognizing causality in verb-noun pairs via noun and verb semantics. *EACL 2014* p. 48 (2014)
34. Rink, B., Bejan, C.A., Harabagiu, S.M.: Learning textual graph patterns to detect causal event relations. In: Guesgen, H.W., Murray, R.C. (eds.) *FLAIRS Conference*. AAAI Press (2010)
35. Selfridge, M.: Toward a natural language-based causal model acquisition system. *Applied Artificial Intelligence an International Journal* 3(2-3), 191–212 (1989)
36. Sporleder, C., Lascarides, A.: Using automatically labelled examples to classify rhetorical relations: An assessment. *Nat. Lang. Eng.* 14(3), 369–416 (Jul 2008)
37. Talmy, L.: Semantic causative types. *the grammar of causative constructions*. *Syntax and Semantics* 6 (1976)
38. Torisawa, K.: An unsupervised learning method for associative relationships between verb phrases. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7. Association for Computational Linguistics (2002)
39. Torisawa, K.: Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurrences. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 57–64. *HLT-NAACL '06*, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)

Discourse Connective - A Marker for Identifying Featured Articles in Biological Wikipedia

Sindhuja Gopalan¹, Paolo Rosso² and Sobha Lalitha Devi¹

¹AU-KBC Research Centre, MIT, Anna University, Chromepet, Chennai, India

²PRHLT Research Centre, Universitat Politècnica de València, Spain

{sindhu jagopalan, sobha}@au-kbc.org, proso@dsic.upv.es

Abstract. Wikipedia is a free-content Internet encyclopedia that can be edited by anyone who accesses it. As a result, Wikipedia contains both featured and non-featured articles. Featured articles are high-quality articles and non-featured articles are poor quality articles. Since there is an exponential growth of Wikipedia articles, the need to identify the featured Wikipedia articles has become indispensable so as to provide quality information to the users. As very few attempts have been carried out in the biology domain of English Wikipedia articles, we present our study to automatically measure the information quality in biological Wikipedia articles. Since the coherence shows representational information quality of a text, we have used the discourse connective count measure for our study. We compare this novel measure with two other popular approaches word count measure and explicit document model method that have been successfully applied to the task of quality measurement in Wikipedia articles. We organized the Wikipedia articles into balanced and unbalanced set. The balanced set contains featured and non-featured articles of equal length and the unbalanced set contains randomly selected featured and non-featured articles. The best result for the balanced set is obtained with F-measure of 83.2%, while using Support Vector Machine classifier with 4-gram representation and Term Frequency-Inverse Document Frequency weighting scheme. Meanwhile, the best result for unbalanced corpus is obtained using the discourse connective count measure with an F - measure of 98.06%.

Keywords: Wikipedia articles quality, Document classification, Featured article, Non-featured article, Word count measure, Discourse connective count measure.

1 Introduction

Wikipedia is a web based, free content encyclopedia with openly editable content. Anyone can write or edit these articles. It was created in 2001 and is a multilingual project in 290 languages. Wikipedia articles are classified into various categories based on their quality. The quality of information includes traditional dimensions such as accuracy, consistency, timeliness, completeness, accessibility, objectiveness and relevancy. Over 4500 articles have been designated as featured articles and 22000

articles as good articles by the Wikipedia community. Featured articles are considered to be the best articles. Wikipedia's strength and weakness is that it is open to anyone. Hence it may also contain low quality content. Non-featured articles are low quality articles that are not of good standard. As Wikipedia articles are increasing enormously in size, it is important to classify these articles as featured and non-featured to provide quality information to the users. The document classification task is to assign a document to one or more classes or categories. Currently there are various document classification works being done on Wikipedia articles of general domain. But, very few works are available for biological domain. Hence, in this study, we have focused on automatically identifying the featured biological Wikipedia articles. In order to increase the participation of computational biology researchers in English Wikipedia and other Wikimedia projects, Wiki Project Computational Biology aims at improving and organizing articles on computational biology, bioinformatics, systems biology and related topics. The overall goal of the project is to improve the article quality of Wikipedia articles within the biology domain [16]. We developed our data set by collecting these articles. Previous works, describes various approaches used to automatically assess the quality of Wikipedia articles.

The paper [2] describes two classification approaches categorization and clustering. This paper discussed the advantages of document classification methods for organizing explicit knowledge. [12] demonstrated the feasibility of building an automatic system to assign quality ratings to Wikipedia articles. Their model obtained an accuracy of 74.6%. [7] investigated four different methods for text classification tasks that include Naive Bayes classifier, the nearest neighbor classifier, decision trees and subspace method. They applied these machine learning techniques to seven class yahoo news groups. The best classification accuracy achieved on seven class problem is approximately 83%. [5] in their work has used inductive learning to categorize natural language documents into predefined content categories.

A variety of simple approaches have been used in the past like number of edits, word count measure, fact count, etc., for automatic classification of featured and non-featured Wikipedia articles. [3] in their paper have analyzed a novel set of features for the task of automatic edit category classification. Using a supervised machine learning experiment, they achieved a micro average F-measure of 62% on a corpus of edits from English Wikipedia. [1] proposed a simple metric word count for measuring article quality. They measured the length of the articles in words. [17] offered new metrics for an efficient quality measurement. Their metrics are based on the life cycles of low and high quality articles. The metrics refer to the changes of the persistent and transient contribution throughout the entire life span. These two metrics are used to measure the editing intensity. [6] in their work demonstrated a simple statistical measure, factual density based on facts extracted from web content using open information extraction. They obtained an F-measure of 90.4% on unbalanced corpus. On balanced corpus, they used relational features for categorizing Wikipedia articles into featured/good and non-featured articles. They obtained an F-measure of 86.7%.

The paper [8] presents the authors' first study to automatically assess information quality in Spanish Wikipedia articles. They evaluated the featured article identification as a binary classification task. Their results show that featured article identifica-

tion for Spanish Wikipedia articles can be performed with an F-measure of 81% when the Support Vector Machines (SVMs) algorithm is used. We have used a novel measure the discourse connective count measure for identifying the featured articles and have compared this approach with other two approaches, word count measure and explicit document model method.

Coherence shows the representational informational quality. Discourse analysis is concerned with measurements of cohesion and coherence. Discourse connective connects the overall text and establishes coherence between the sentences and coherence shows how well the information hangs together. It gives completeness and relevance among the text [12]. Since the quality of the featured article lies on these traditional dimensions, connectives can be used as a measure to distinguish featured and non-featured articles. In past works various algorithms have been proposed to measure cohesion. However, coherence is more difficult to quantify. We have used discourse connectives as a measure of coherence. We have collected the featured and non-featured Wikipedia articles from the index of biology articles that belongs to general biology, molecular biology and evolutionary biology. Then it is organized into balanced and unbalanced set. First, we used word count measure to classify the articles. Then, we used the discourse connective count measure. The results obtained after using connectives showed that this feature outperforms the word count measure for unbalanced corpus. On balanced corpus, the explicit language model method performed better than the word count measure and discourse connective count measure. The results are comparable with state-of-art systems.

In the next section, we describe the datasets used to develop the balanced and unbalanced corpus. In Section 3 experiments performed are explained in detail and in Section 4 results obtained are discussed. We conclude our paper in Section 5.

2 Corpus Used

Our data set consists of totally 2028 featured and non-featured biological Wikipedia articles. The Wikipedia article quality grading scheme classifies articles into different classes. In our work we have considered the Wikipedia articles belonging to Featured article, A, Good Article, B and B plus categories as featured articles and articles belonging to C, start and stub class as non-featured articles. Featured articles are well written, accurate, and stable and images are well illustrated. These articles are well organized and complete. The non-featured articles miss important content and contain irrelevant information. These featured and non-featured articles are organized as the balanced corpus and the unbalanced corpus. The balanced corpus contains the featured and non-featured articles of similar length. The unbalanced corpus contains randomly selected featured and non-featured articles without considering the document size. The balanced and unbalanced corpora contain 518 featured and 496 non-featured articles each. For the experiments on the balanced and the unbalanced corpora, we have used 811 articles for training and 203 articles for testing in the ratio 80:20. Figure 1 shows the corpus statistics.

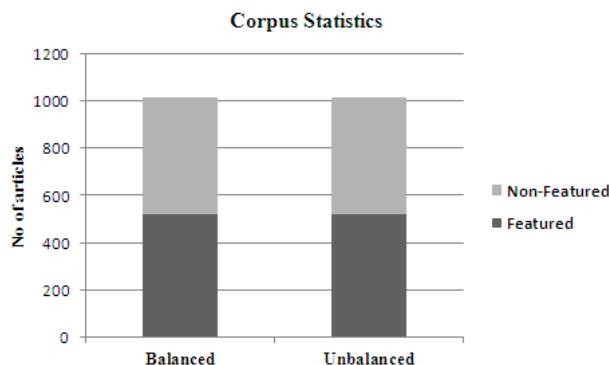


Fig. 1. Corpus statistics

3 Experiments

In our work, we automatically classify the biological Wikipedia articles as featured and non-featured articles. To identify the biological featured articles, we used word count measure, connective count measure and explicit document model method. These approaches are described in detail in the following sections.

3.1 Word Count Measure

First, we evaluated the word count measure as a baseline on the balanced and unbalanced corpora. The word count measure is a simple and effective measure of quality for Wikipedia articles [1]. It simply means the number of words in a document. Based on the number of words, each article is classified as featured and non-featured. To evaluate using the word count measure, first the documents were tokenized and the numbers of words were counted. While, [1] in his work have classified the articles with more than 2000 words as featured and those with fewer than 2000 words as random, [6] have used word count of 200 for lower quality articles and 1400 for high quality articles. [8] used a threshold of 3070 words for the unbalanced set and 955 words for the balanced set to classify featured and non-featured articles for Spanish Wikipedia articles. In this study, we found that the word count of non-featured articles on unbalanced corpus is lesser than 1800 words. Meanwhile, on the balanced corpus the word count of non-featured articles is between 800 to 3600 words and word count of featured articles is between 2700 to 15000 words. Hence, on an average the unbalanced corpus articles having more than 2000 words and balanced corpus articles having more than 3000 words are categorized as featured articles.

3.2 Discourse Connectives Count Measure

The quality of information lies in how well the provided information is useful to the users. [10], has described ten dimensions of information quality. The quality infor-

mation has to meet certain criteria like accuracy, timeliness, relevancy, etc. coherence is one among the ten dimensions that plays a significant role in defining information quality. Coherence makes a text semantically meaningful. It can be achieved through syntactic features such as deitic, anaphoric and cataphoric elements, presuppositions etc. [13]. Discourse connectives are one such syntactic feature that establishes coherence between two units in a text/discourse. They connect two discourse units that include single clauses or multiple clauses and in some cases it may include whole sentences and even multiple sentences. The units the discourse connectives connect are called as arguments. The relation can be established explicitly or implicitly [11].

Example 1

Some DNA sequences are transcribed into RNA **but** they are not translated into protein products.

In Example 1, “but” is the explicit discourse connective that connect two clauses. Here, connective “but” establishes coherence between two clauses.

Example 2

In the absence of SOX2, there is no equivalent rapidly proliferating cell population, the only surviving cells being trophoblast giant cells **and** ExEn.

In Example 2, “and” acts as a connective that connect two entities “trophoblast giant cells” and “ExEn”. Here “and” is not a discourse connective because the minimal unit required for a connective to act as discourse connective is a clause that is tensed or non-tensed [11].

Example 3

Further studies found that L-PHP was expressed in pancreas. **<IMPLICIT: However>** The biological role of pancreatic L-PHP is still not clear.

In Example 3, the two sentences are related but there is no discourse marker that explicitly shows the relation. Hence the relation can be established implicitly by inserting a discourse connective “however”.

In our work, we have considered explicit discourse connective count measure to identify the featured articles in biology domain. The discourse connectives are first identified from the text and the connective count is obtained. Since all the connectives in a text are not discourse connectives as in Example 2, it is necessary to develop a system to automatically identify the discourse connectives. We followed a similar method used by [4], to develop a system for automatic identification of discourse connectives. They have used a hybrid approach using linguistic rules and machine learning approach to identify the discourse relations. Likewise, we used the CRF++ tool [14], an open source implementation of Conditional Random Fields (CRFs) and linguistic rules to develop the system. PubMed abstracts were tagged with discourse connectives. Then the documents were tokenized and the features were extracted. Word, Part of Speech (PoS), Chunk, Combination of word, PoS and chunk were used as features. This corpus is trained using CRF++ tool and language models are created. Further, we also used linguistic rules to identify the connectives. The accuracy of the system is 97.3%.

Using this system the discourse connectives are identified automatically from the Wikipedia articles. The word count directly influences the discourse connective count, i.e. if the number of words in a document is higher, then number of connectives will be higher. In this dataset, the non-featured articles on the unbalanced corpus contain less than 50 connectives on an average, while the connective count of non-featured articles on the balanced corpus is between 50-270 connectives. Therefore, we performed our experiments by setting an average threshold of 50 connectives on the unbalanced corpus and 150 connectives on the balanced corpus for featured articles. Figure 2 shows the discourse connective count in the unbalanced corpus and Figure 3 shows the discourse connective count in the balanced corpus.

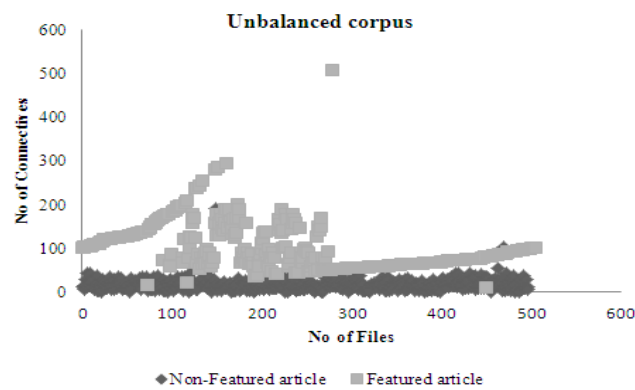


Fig. 2. Discourse connective count in the Unbalanced Corpus

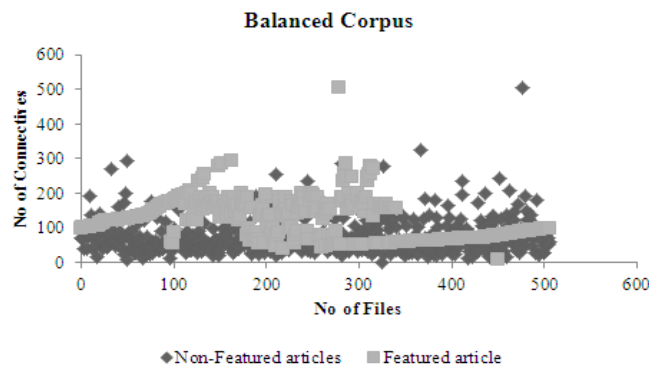


Fig. 3. Discourse connective count in the Balanced Corpus

We finally used explicit document model method to classify featured and non-featured articles for both the balanced and unbalanced corpora, which is described in detail in the next section.

3.3 Explicit Document Model

Finally, we employed explicit document model method [7] to identify featured articles from biological Wikipedia articles. [8] used this method for Spanish Wikipedia document classification. Their corpus includes the articles that belong to the snapshot of the Spanish Wikipedia from 8th July, 2013. They used two classifiers, Naive Bayes (NB) and SVMs with Term Frequency-Inverse Document Frequency (TF-IDF) and binary document models for the balanced and unbalanced corpora. We applied this method for classification of biology Wikipedia articles into featured and non-featured articles. We performed the experiments using WEKA data mining software [15] and used NB and LIBSVM classifiers. Explicit document model representation includes n-gram vectors and bag-of-words.

We extracted 3, 4 and 5 grams from plain text. Bag-of-words is a simple representation used in Natural Language Processing and Information Retrieval. It is commonly used in the methods of document classification, where the occurrence of each word is used as feature for training a classifier. In our work, n-gram vector and bag-of words are used as features with TF-IDF and binary weighting schemes.

TF-IDF is a numerical statistics. It shows the importance of a word in a document or corpus. It is the product of term frequency (TF) and inverse document frequency (IDF). It is a way to score the importance of words in a document based on how frequently they appear across multiple documents. TF is the number of times a word appears in a document normalized by dividing the total number of words in a document.

TF(t) = Number of times term t appears in a document / Total number of terms in the document. (1)

IDF measures how common a word is among all documents. An inverse document frequency factor diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

IDF(t) = \log_e (Total number of documents / Number of documents with term t in it). (2)

TF-IDF is simply the product of TF(t) and IDF (t).

TF-IDF= TF(t) * IDF(t) (3)

Words with high TF-IDF suggest that if the word appears in a query, the document could be of interest to the user. We used the information gain method to rank the words to be used as features. For bag of words representation, both binary document model and TF-IDF weighting scheme had words connective, fossil, gel, flowers, ecology, heart, mRNA, intermediates and host ranked in first 30 positions. The results obtained are discussed in the next section.

4 Results and Discussion

First, we used the word count measure to classify the featured and non-featured articles. We obtained an F-measure of 92.47% on the unbalanced corpus. For the balanced corpus, we obtained an F-measure of 56.1%. This method showed good performance on the unbalanced corpus, while on the balanced corpus word count measure failed to give good results. Then, using the novel measure discourse connective count measure we performed our experiments on the balanced and unbalanced corpora. For the unbalanced set, we obtained an F-measure of 98.06%, when the threshold was set to 50 connectives. For the balanced set, we obtained 65.61% F-measure, when the threshold was set to 150 connectives. The discourse connective count measure outperformed the word count measure on unbalanced corpus. For the balanced corpus, the discourse connective count measure performed better than the word count measure. Finally, we performed our experiments using the explicit document model method.

The results for NB and SVMs classifiers with TF-IDF and binary document models on the balanced and unbalanced corpora are presented in the Table 1 and 2.

Table 1. F-measure in % for featured and non-featured articles classification on the balanced corpus

Features	Balanced Corpus			
	Binary		TF-IDF	
	NB	SVM	NB	SVM
Bag of words	79.6	81	79.1	79
3-grams	75.7	81	73.6	79.5
4-grams	78.6	82.9	81.5	83.2
5-grams	76.6	80.5	79.1	79.5

Table 2. F-measure in % for featured and non-featured articles classification on the unbalanced corpus

Features	Unbalanced corpus			
	Binary		TF-IDF	
	NB	SVM	NB	SVM
Bag of words	94.6	95.1	94.6	95.4
3-grams	96	97	92.6	96
4-grams	95.1	95.6	91.7	97.1
5-grams	96	96.5	94.1	95

The above results show that SVM performs slightly better than the NB method. The best result on the balanced and the unbalanced corpora is obtained using SVM classifier with 4-gram representation and TF-IDF weighting scheme. We obtained F-

measure of 83.2% on the balanced corpus and 97.1% on the unbalanced corpus. The system shows better performance on the unbalanced corpus than the balanced corpus. However, this result on unbalanced corpus shows that the discourse connective count measure performs better than explicit document model method. [6] used fact frequency based features and obtained an F-measure of 90.4% on unbalanced corpus. Fact frequency based features require direct information about the number of facts obtained by an information extraction process from a text. The facts are computed using Reverb Open Information extraction framework.

Using the fact count factual density is calculated. This feature worked well for identification of featured articles on unbalanced corpus. [8] used the word count discrimination rule and obtained an F-measure of 96% for unbalanced corpus for Spanish Wikipedia articles. [1] achieved 96.31% on an unbalanced corpus of English Wikipedia articles using the word count measure. [9] have identified featured articles from English Wikipedia domains biology and history. Their unbalanced set contained featured and non-featured articles in the ratio 1:6 respectively. They used word discrimination rule and obtained accuracy of 96% on unbalanced corpus. From the results of previous works, our works show that the discourse connective count measure outperformed word discrimination rule and factual density measure for unbalanced corpus. On the balanced corpus, the discourse connective count measure showed better performance than word count measure, but the explicit document model approach outperformed both the approaches.

The best result on balanced corpus is obtained using SVM method with F-measure of 83.2%. [8] achieved highest F-measure of 80% for 4-grams features using SVM classifiers when applied to binary representation. [9] obtained an accuracy of 96% within Biology and 92% within History when the binarized character trigram vector representation combined with an SVM was used. [6] used relational features to classify Wikipedia articles into featured/good and non-featured ones. For articles of similar lengths, they achieve an F-measure of 86.7% and 84% otherwise. The results show that the explicit document model method showed “comparable” state-of-art results on balanced corpus.

5 Conclusion

In our work we carried out various experiments to automatically classify the featured and non-featured biological Wikipedia articles. We created two corpora, balanced and unbalanced. The balanced corpus contains featured and non-featured articles of equal length, while the unbalanced corpus contains articles of dissimilar length. The word count measure, discourse connective count measure and explicit document model approach were used to identify the featured articles. The word count measure is a simple method used in the past for identifying the featured articles. Hence we used this method as a baseline to categorize the biological Wikipedia articles. Then, we used a novel approach, the discourse connective count measure. This measure outperformed other approaches used in the past work on the unbalanced corpus. Finally, we used machine learning classifier NB and SVM on the balanced and unbalanced corpo-

ra. We used bag of words and n-gram features with TF-IDF and binary weighting schemes. N-gram features include 3-gram, 4-gram and 5-gram representation. We obtained best results using SVM classifier and 4-gram feature. The results obtained on the balanced and unbalanced corpora are “comparable” to state-of-art systems.

Acknowledgement. This work is the result of the collaboration between AU-KBC Research Centre, Chennai, India and the Universitat Politècnica de València (UPV), Spain in the framework of the WIQ-EI IRSES research project (grant no. 269180) within the EC FP7 Marie Curie. The work of the second author is also in the framework of the SomEMBED TIN2015-71147-C2-1-P MINECO research project and by the Generalitat Valenciana under the grant ALMAPATER (PrometeoII/2014/030).

References

1. Blumenstock, J.: Size Matters: Word Count as a Measure of Quality on Wikipedia. In: 17th International Conference on World Wide Web, pp. 1095–1096. Beijing, China (2008)
2. Brucher, H., Knolmayer, G., AndreMittermayer, A.: Document Classification Methods for Organizing Explicit Knowledge. In: Third European Conference on Organizational Knowledge, Learning, and Capabilities, Athens, Greece(2002)
3. Daxenberger, J., Gurevych, I.: Automatically Classifying Edit Categories in Wikipedia Revisions. In: Conference on Empirical Methods in Natural Language Processing, pp. 537-547. Seattle, Washington, USA (2013)
4. Lalitha Devi, S., Gopalan, S., Sreedhar. L., Rao, P.R.K., Ram, R.V.S., and Malarkodi C.S.: A Hybrid Discourse Relation Parser in CoNLL 2015. In: Nineteenth Conference on Computational Natural Language Learning: Shared Task, pp. 50-55. Beijing, China (2015)
5. Lewis D David., Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. In: Third Annual Symposium on Document Analysis and Information Retrieval, pp. 81-93. ISRI, Las Vegas(1994)
6. Lex, E., Volske, M., Errecalde, M., Ferretti, E., Cagnina, L., Horn, C., Stein, B., Granitzer, M.: Measuring the Quality of Web Content Using Factual Information. In: 2nd Joint WICOW/AIRWeb Workshop on Web Quality, pp. 7-10. New York, USA (2012)
7. Li, Y.H., Jain, A.K.: Classification of Text Documents. *The Computer Journal*. 41(8), 537-547 (1998)
8. Lian, P., Edgardo, F., Errecalde, M.: Identifying Featured Articles in Spanish Wikipedia. In: Feierherd, G.E., Pesado, P.M., Spositto, O.M. (eds.) XX Argentine Congress of Computer Science Selected Papers. Computer Science & Technology Series, pp. 171-182. (2015)
9. Lipka, N., Stein, B.: Identifying Featured Articles in Wikipedia: Writing Style Matters. In: 19th International Conference on World Wide Web, pp. 1147–1148. Raleigh, USA (2010)
10. Miller, H.: The Multiple Dimensions of Information Quality. *Information Systems Management*. 13(2), 79-82 (1996)
11. Prasad R., Dinesh N., Lee A., Miltsakaki E., Robaldo L., Joshi A., Webber B.: The Penn Discourse Treebank 2.0. In: Sixth International Conference on Language Resources and Evaluation (LREC'08), pp. 2961-2968. Marrakech, Morocco.(2008)
12. Rassbach, L., Pincock, T., Mingus, B.: Exploring the Feasibility of Automatically Rating Online Article Quality. In: 9th Joint Conference on Digital Libraries, (2007)

13. Rouchota, V.: Discourse Connectives: What do They Link?. UCL Working Papers in Linguistics. 8, 199-214 (1996)
14. Taku, K., CRF++, an Open Source Toolkit for CRF, <http://crfpp.sourceforge.net>,(2005)
15. Waikato Environment for Knowledge Analysis, http://www.iasri.res.in/ebook/win_school_aa/notes/WEKA.pdf
16. Wikipedia: WikiProject Computational Biology, en.wikipedia.org/wiki/Wikipedia:WikiProject_Computational_Biology
17. Wohnner, T., Peters, R.: Assessing the Quality of Wikipedia Articles with Lifecycle Based Metrics. In:5th International Symposium on Wikis and Open Collaboration, pp. 1-10. Orlando, Florida, USA(2009)

Named Entity Recognition for the Agricultural Domain

Malarkodi C. S.¹, Elisabeth Lex², Sobha Lalitha Devi¹

¹AU-KBC Research Center, MIT Campus of Anna University, Chromepet, Chennai, India

²Knowledge Technologies Institute, Graz University of Technology, Austria
{csmalarkodi,sobha}@au-kbc.org, elex@know-center.at

Abstract. Agricultural data have a major role in the planning and success of rural development activities. Agriculturalists, planners, policy makers, government officials, farmers and researchers require relevant information to trigger decision making processes. This paper presents our approach towards extracting named entities from real-world agricultural data from different areas of agriculture using Conditional Random Fields (CRFs). Specifically, we have created a Named Entity tagset consisting of 19 fine grained tags. To the best of our knowledge, there is no specific tag set and annotated corpus available for the agricultural domain. We have performed several experiments using different combination of features and obtained encouraging results. Most of the issues observed in an error analysis have been addressed by post-processing heuristic rules, which resulted in a significant improvement of our system's accuracy.

Keywords: Named Entity Recognition, Text mining, Information Extraction, Natural Language Processing, Agricultural data

1 Introduction

Named Entity Recognition (NER) has a fundamental role in Information Extraction (IE) and text mining applications like Question/Answer (Q&A) systems, event/product monitoring or customer-product relation extraction systems. For instance, in Q&A systems, named entities are answer strings to 'WH' questions. Text mining attempts to find the knowledgeable information from unstructured data lying on the web [24]. IE is considered as an important component of Text mining, since it aims to represent the structured information from the unstructured web data. The structured information extracted from IE systems, are mainly named entities. Since the identification of named entities discovers the hidden knowledge from data, it is considered as the main component in various activities related to text mining. The main goal of this work is to develop NER system for agriculture domain which supports the Information Systems and Text mining activities concerning agriculture sector. Agriculture is a main source of livelihood. Information about agricultural sciences are essential to improve agricultural productivity, research & development efforts. Food and agriculture organization emphasize that agricultural information is the cru-

cial factor in rural development activities [21]. For example, stake-holders within the agricultural domain, such as livestock & food processing industries, pesticide industries, researchers, or policy-makers need to retrieve entities such as crop names, fertilizers and price factors and how they are reflected for sound decision making regarding agricultural products. Though the extensive amount of work has been done on NER across various domains, very few have been found in the agricultural domain. NER from agricultural data enlightens the following information; 1) crop production and its location 2) policies or schemes benefiting farmers 3) natural disasters affecting crop cultivation 4) pesticides to control the pests 5) diseases affecting plant growth. The present work focused on the developmet of NER system for agriculture domain. We have developed the fine-grained NE tagset with 19 tags which covers the main key-terms in the respective domain. The paper is organized as follows: section 1 presents the overview of the state-of-art systems about NER, Section 2 describes the tagset design, corpus collection and NE annotation, Section 3 explains the various features used for system development, Experiments, results and error analysis are explained in Section 4. Finally the paper concludes with section 5.

1.1 Related Work

NER aims at the identification and classification of proper nouns into predefined categories like person, location, organization, etc.. Initially, NER was defined in MUC 6 as part of Information Extraction (IE). A survey of fifteen years of research on NER has been conducted by Nadeau [17]. They presented various features and methods used for NE identification from 1996 to 2006. A high performance named entity system for English and Spanish had been built using the standard version of a Hidden Markov Model (HMM) and obtained 90% accuracy [3]. Another NER system was developed to extract person names from e-mail[16]. Over a couple of decades, lot of research has been carried out on NER across languages. For example, Rössler et al. developed a NE engine for German based on Support Vector Machines (SVM) [19]. A NER system has been built for Dutch based on a genetic algorithm approach[5], an another for English using a combination of four classifier [8], while Federico et al. proposed a NER for Italian that is based on a boot-strapping process [7].

A more general overview on approaches used for NER across laguages is given in the survey of Kaur et al. [9]. They listed NER methods that follow rule based approaches, machine learning techniques such as decision trees, Naïve Bayes, Hidden Markov Model (HMM), Maximum Entropy Model (MEMM), Conditional Random Fields (CRF) and hybrid approaches, i.e. the combination of both rule based and machine learning approaches.

Vijayakrishna et al. worked on Tamil NER for the Tourism domain using CRF. Their system handles nested tagging of named entities with a hierarchical tag set containing 106 tags [22]. Malarkodi et al. proposed a NER model with language dependent and independent features for English, Tamil, Telugu, Bengali, Punjabi and Marathi using CRF [15]. Ekbal et al. developed NER systems for the two leading Indian

languages, namely Bengali and Hindi using CRF [6]. Bindu et al. developed the Malayalam NER engine and discussed how their work support Q&A systems [4].

The work presented in [18] is also very much related to ours as they also perform NER in the domain of agriculture. In their work, they constructed auto-matic NE gazetteers using unsupervised learning, more specifically, a variant of Multiword Expression Distance. They have used three NE tags, namely crop, disease and chemical_treatment. For each NE type, the gazetteers were generated automatically and the effectiveness of the dictionary was compared with Wikipedia articles related to agriculture.

1.2 Our Contributions

The contributions presented in this work are threefold: 1) In order to extract a wide range of information from the agriculture domain, we have designed a fine-grained tag-set comprising of 19 entity types. 2) The NE annotated corpus has been created for agriculture with 1L word forms 3) We have constructed the baseline system with basic minimal features such as word, POS and chunking information 4) Based on the analysis, we also incorporated rich set of linguistic features for the system development 5) Post processing heuristics is applied further to fine-tune the system.

2 Tagset Design and Corpus Preparation

2.1 Our NER Tagset

We have paid special attention towards developing fine-grained tagsets for our work. The proposed tagset consisting of 19 tags for named entities can help to understand the semantic classes of entities for the agricultural domain. Table 1 explains the rationale behind the tags along examples.

Table 1. NE Tagset With Example

NE Tags	Description	Example
Person	Names of person and nationals	John Smith, Indian
Location	Names of cities, continents, water bodies	Europe, Chennai
Organization	Names of institutions, companies, industries	Common organization Of agricultural market
Chemicals	Fertilizers, pesticides, fungicides	Nitrogen, nitrate
Crop	Names of fruits, vegetables, cereals, grains	Apple, carrot, wheat
Organism	Names of animals and micro-organisms	Sheep, Escherichia coli,
Policy	Agricultural aids or policies	Common agricultural Policy
Climate	Denotes the climatic conditions	Summer, winter

Food items	Plant/animal products	Cheese, milk, bread
Diseases	Diseases affecting plant growth	Late Blight, brown rot
Natural Disaster	Disasters affecting crop production.	Famine, earthquake, Flood
Events	Conferences, workshops, meetings and exhibition	National conference on agricultural & Food Security
Nutrients	Fats, minerals, vitamins and carbohydrates	Vitamin A, calcium
Count	Number of items	350 people
Distance	Distance measures such as feet, meter, km., etc.	15 inches, 250 acres
Quantity	Quantity measures such as litter, tonnes, grams	8 tonnes, 10 kg
Money	Currency value such as the euro, rupee, dollar etc.	\$90, 100 euro, Rs. 1000
Temperature	Numerical measure of climatic condition such as degree, Celsius	70C
Year_Month_Date	Denotes the year, month, day and date	May 26, 29/10/2013

2.2 Corpus Collection and Annotation

Corpus Collection. As the system's performance depends on training data for the CRF, the corpus collection is a crucial factor. The dataset used in the proposed work has been collected from Wikipedia articles that are related to agriculture as well as from reputed websites in the European Union pertained to agriculture. The corpus has been collected in such a manner that it would cover major aspects of farming from crop cultivation to agricultural productivity. Therefore for data collection we focused on the sub-domains like crop cultivation and management, food processing industries and research institutions, subsidiaries, organisms and diseases, food products, natural disasters and risk management, agribusiness and marketing.

NE Annotation. The agricultural corpus comprises of 100k word forms. The raw text is tokenized and pre-processed with part of speech and chunking information. Named entities are manually annotated and represented in BIO standard mentioned in CONLL 2003 shared task. Issues encountered during the annotation process are as follows: 1) ambiguity between NEs 2) ambiguity between NE and non-NE 3) boundary limitation 4) abbreviations. Firstly, there exists an ambiguity between named entities of one type and another; for example, Jersey is a location name in some instances and Jersey is a cattle name in some cases. Purcari is a wine sector name in some examples and the same instance denotes the wine name in other examples. Secondly, ambiguity aroused in the case of whether to consider certain instances as NE or not. For example, "common organization of agricultural markets" is an organization name but ambiguity arisen whether to consider words like fast-food market, agricultural market, as NE or non-NE. As those instances are important in this domain, we con-

sidered them as NE. Thirdly, due to the definite descriptions it is difficult to determine the NE boundaries. Fourthly, abbreviations of organization, policy and event names are hard to identify without contextual knowledge. The corpus statistics are shown in the table 2. The corpus is randomly divided into 80:20 for training and test data.

Table 2. Corpus Statistics

Corpus	Total No of Words	Total No of NEs
Train Corpus	80k tokens	8,924
Test Corpus	20k tokens	2,117

3 Our Approach

CRF is a probabilistic model used to segment and label the sequential data [12][23]. It selects the label sequence y which maximizes the conditional probability of $p(y|x)$ to the observation sequence x . The probability of a label sequence y given an observation sequence x is given below:

$$P(y|x, \lambda) = \frac{1}{z(x)} \exp \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}, y_i, x, i)$$

$$z(x) = \sum_{i=1}^n \sum_j \lambda_j f_j(y_{i-1}, y_i, x, i)$$

Where z is normalization factor, $f_j(Y_{i-1}, Y_i, x, i)$ is a transition feature function of an observation sequence and the labels at position i and $i - 1$. For example, consider the task of assigning the label y to the word x named 'Netherlands', then the transition function $f_j(y_{i-1}, y_i, x, i) = 1$ if $y_i = \text{"LOCATION"}$ and the suffix of i_{th} word is "land"; otherwise 0; If the weight λ_j associated with the above feature is large and positive, then the words ending with the suffix "land" is labelled as NE type "LOCATION".

We have developed the base NER engine, with minimal basic features. Later, we added heuristic rules to improve the system's performance. Our method is a hybrid approach, as we are using both machine learning technique and linguistic rules. We came up with the linguistically inspired features which are explained below.

3.1 Syntactic Level Features

POS and Chunk Information. POS play a significant role in NE identification, as they contain information about the linguistic category of words. We have considered the POS tags occurring in a window of five. Noun phrase chunking helps identify the NE boundary.

Proper Noun. As most of the Named Entities belong to the grammatical category proper noun, we gave importance to the POS tag of the proper nouns.

Frequent POS patterns of NE. The most frequent POS tags preceding the NE occurring in w-1(word preceding the NE) and w-2 positions are considered as a feature.

3.2 Lexico-Syntactic Features

Cue Phrases and POS. Cue phrases are the key words occurring as part of an entity. For instance, organization names are following or preceding by cue phrases like university, consortium, ltd. The key-terms like blight, rots are occurring as part of plant disease.

Occurrence of proper noun after preposition 'in'. We have examined the proper nouns following the preposition 'in'. Our analysis revealed that in most of the cases proper noun (NNP) that comes immediately after the preposition 'in' are location names.

Numerical Feature. Digit patterns exhibit useful information in predicting numerical entities. For instance, numerical values of length four tend to identify year names. POS of the numerical value CD (Cardinal Number) preceded and succeeded by quantity measures and distance measure are considered as a feature.

4 Experiments and Results

The agriculture corpus is randomly divided in the ratio of 80:20 for training and test data. We measured the performance of our system in terms of precision, recall and f-measure. In order to find the best feature set, we have conducted several experiments using various combinations of features. We also have performed 10 fold cross validation. Feature-wise performances are shown in tTable 3.

Table 3. Feature-wise Results

S.NO	Feature Combination	Precision	Recall	F-M
1	Word, POS and chunk information	80.96	71.75	76.35
2	(1), capitalization	81.59	72.63	77.11
3	(1), noun phrase	81.69	72.63	77.16
4	(1),(2),(3), Occurrence of NP after 'IN'	82.67	74.15	78.41
5	(1),(2),(3),(4), key words	82.83	76.01	79.42
6	(1),(2),(3),(4),(5), affixes	83.50	77.58	80.54
7	(1),(2),(3),(4),(5),(6), Numeric features	83.61	78.81	81.21
8	(1),(2),(3),(4),(5),(6),(7),POS patterns of NE	84.45	79.67	82.06

4.1 Contribution of Various Features

Initially, we have applied the combination of word, POS and chunking information to determine the performance of our system when using the minimal feature set. We obtained an accuracy of F-M: 76.35% for the base NE system. In comparison with

results shown in a row (1), including capitalization as feature improves the accuracy by 1% and including proper nouns leads to the increase of 1% in precision and 1% in recall value. The occurrence of proper noun after the preposition “in” enables a positive effect on f-measure by 1%. Especially this feature increased the performance of NE type “Location”, since most of the proper nouns followed by the preposition “in” are location names. The key word feature provides the improvement of 2% recall while the affix feature provides a slight improvement on f-measure. Inclusion of keyword feature contributes to raise an accuracy of NE types such as “organization, policy, diseases”. The affix feature leads to the improvement of chemical and location entities. Finally, the incorporation of numerical features boost the identification of numerical entities such as “Unit, Money and Year_Month_Date” and POS patterns succeeding & preceding NE seems to be effective for all NE types. The highest accuracy obtained from the CRF approach is 82.06% (F-Measure). Tag-wise, F-Measure (F-M) values for entity expressions are depicted in Figure 1.

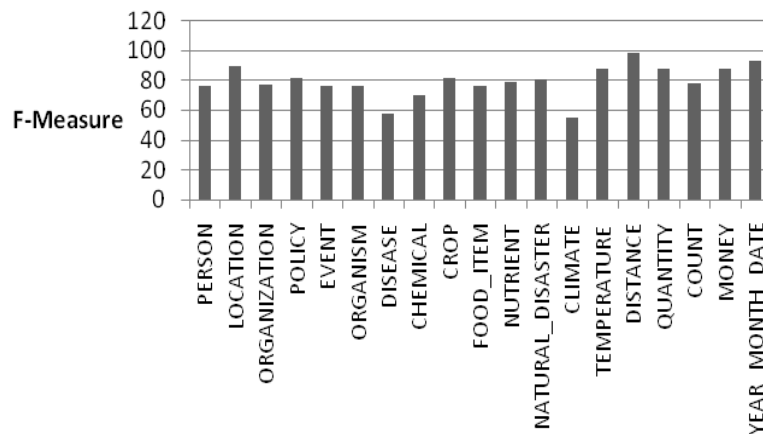


Fig. 1. Tag-wise Results from CRF

4.2 Cross Validation

Cross validation is used to assess how our results can be generalized to an independent dataset. Specifically, we performed 10 fold-cross validation and report the average accuracy is shown in table 4. During the analysis on classification results, we have observed that 70 percent of Organization names in our corpus consist of more than six token length and 80% of such instances are correctly identified by the system. In entity expressions, location tag obtained the highest score of 90.83% precision and 88.19% recall. Figure 1. shows that other than the entities DISEASE and CLIMATE, all entity expressions scored above 75% accuracy. Except NE type COUNT, we achieved more than 80% for all numeric entities. Crop name was the second highest one.

Table 4. 10-Fold Cross-Validation Results

S. No.	Precision	Recall	F-M
1	83.06	77.48	80.27
2	84.20	78.25	81.22
3	84.60	79.13	81.86
4	84.73	81.35	83.04
5	82.05	79.06	80.55
6	85.55	82.22	83.88
7	82.30	78.66	80.48
8	86.95	82.05	84.50
9	81.81	77.21	79.51
10	87.32	80.84	84.08
Avg	84.25	79.62	82.23

4.3 Error Analysis

From our observation, we found that the errors in our system are due to inconsistencies and ambiguities between entities. Abbreviation of policy names is misclassified as organization in some cases. Such instances can be handled using the NE type of the respective expansion. In some cases, abbreviations will occur right after the expansion. In the example of the Common Agricultural Policy (CAP), if the system has tagged the expansion correctly and fails to identify the abbreviation 'CAP', we can tag the NE type for 'CAP' as 'POLICY' by using the NE tag of the expansion which precedes the abbreviation. Parts of organization names may be misclassified as Location and food_item as crop type in a few instances. This problem arises when one type of NE occurs as part of another type of NE (entity within entity, i.e. a so-called nested entity). It can be handled using the cue phrases of the respective types with the combination of POS and orthographic features.

4.4 Impact of Post-processing

In order to improve the accuracy of our system and to remove inconsistencies, we applied linguistic and heuristic rules following the CRF output. Some of the rules implemented in the post-processing are discussed below.

The rule 1 given in table 5 describes that if the B-tag (Beginning Tag) of same NE type occurs in two consecutive positions within the same phrase, the second B-tag should be replaced as I-tag (Inside Tag). The rule2 illustrates that if the I-tag start without B-tag, then the I-tag in the beginning of an entity should be changed to B-tag. In row3, instead of tagging “Sugar Cane Juice“ as food_item, the system has tagged part of the NE “Sugar Cane“ as “CROP“. This ambiguity has occurred due to being a nested entity. As we considered the maximal entity, we handled this ambiguity using linguistics rules based on the POS, orthographic features and key words. Thus, we have handled partial tagging and ambiguities that exists between NEs. Tag wise results obtained after post-processing is given in Table 6.

Table 5. Post Processing Results

Post Processing Rule	Example	Comments
-2 NNP+B-NP B-TAG1	-2 John B-PER	-2 John B-PER
-1 NNP+I-NP B-TAG1	-1 Mathew B-PER	-1 Mathew I-PER
0 NNP+I-NP I-TAG1	0 Zen I-PER	0 Zen I-PER
-2 NNP I-TAG1	2 Rural I-POLICY	-2 Rural B-POLICY
-1 NNP I-TAG1	-1 Agri I-POLICY	-1 Agri I-POLICY
0 NNP I-TAG1	0 Policy I-POLICY	0 Policy I-POLICY
0 NNP B-TAG1	0 Sugar B-CROP	0 Sugar B-FOOD_ITEM
1 NNP I-TAG1	1 Cane I-CROP	1 Cane I-FOOD_ITEM
2 Keyword+NN NNP	2 Juice O	2 Juice I-FOOD_ITEM

Table 6. Tag-wise results after Post-Processing

NE Type	Precision	Recall	F-measure
PER_LOC_ORG	82.59	83.59	83.09
CROP	82.31	76.7	79.50
ORGANISM	84.28	74.68	79.48
DISEASE	62.5	98.5	80.50
CHEMICAL	86.27	72.13	79.20
FOOD_ITEM	80.32	78.01	79.16
POLICY	82.17	81.1	81.63
OTHERS	85.93	83.26	84.59
\NUMERICAL	91.50	90.43	90.96
TIME	92.54	92.95	92.74
AVERAGE	83.24	83.13	83.18

Where PER_LOC_ORG- Person, Location and Organization, OTHERS include the NE type Natural Disaster, Climate, Nutrient and Event. As the post processing rules reduced the number of false positives and increased the number of true positives, there was a slight decrement in precision and 4% increment in the recall. Figures 2 and 3 show the results before and after post-processing.

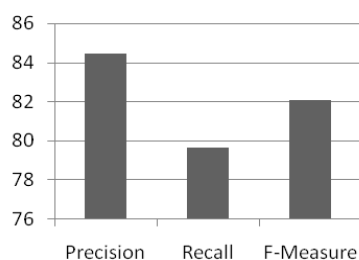


Fig. 2. Results before Post-Processing

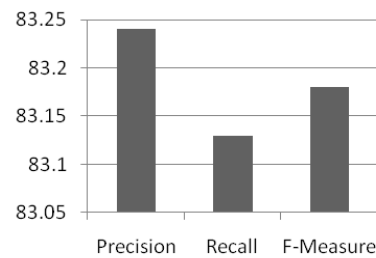


Fig. 3. Results after Post-Processing

4.5 Comparison with CreateGazetteMED

Patil et al. worked on the agriculture domain with 3 NE tags[18]. The highest precision obtained by the CreateGazetteMED algorithm is 66.2% for crop, 92.8% for disease and 88.6% for chemical. We have achieved a precision of 82.31% for crop, 62.5% for disease and 86.27% for chemical. The precision we scored for the NE category crop is higher and chemical is quite closer to CreateGazetteMED results. There was a precision drop for disease tag, due to the false positives.

5 Conclusion

In this paper, we have presented a NER system for the agriculture domain. To the best of our knowledge, this work is the first attempt in generating a NE annotated corpus and NE system with a major tagset for agriculture. We have collected data from various sub-domains of agriculture starting from cultivation to marketing and we designed the NE tag-set with 19 fine grained tags so that it could cover prominent entities in the agricultural field. Our system exploits both linguistically enriched as well as domain independent features. With our system, we achieved 76% accuracy with minimal features such as word, POS and chunk information. Based on a detailed corpus analysis, more features are incorporated and our results were improved by 7%. We have implemented post-processing heuristics to overcome tag ambiguities which resulted in an improvement of the overall precision of our system to 83.24% and a recall of 83.13%. Our results show that our system is comparable with existing NER models. In future we plan to extend this work by developing a robust relation extraction system for the agricultural domain.

Acknowledgement

This work has been funded by the European Commission as part of the WIQ-EI project (project no. 269180) within the FP7 People Programme. It is the result of the collaboration between AU-KBC Research Centre, Chennai, India and Know Center, Graz, Austria. The Know-Center is funded within the Austrian COMET program – Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth, and the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Palanisamy, A., Lalitha Devi, S.: HMM based POS Tagger for a Relatively Free Word Order Language. *Research in Computing Science*, vol. 18, pp. 37-48. (2006)

2. Benajiba, Y., Rosso, P.: Arabic named entity recognition using conditional random fields. In: Proceedings of the workshop on HLT & NLP within the Arabic World, LREC, vol. 8, pp. 143-153. (2008)
3. Bikel, D.M., Miller, S., Schwartz, R., Weischedel R.: Nymble:A high-performance learning name-finder. In: Proceedings of Fifth Conference on Applied Natural Language Processing, pp. 194-201. (1997)
4. Bindu, M.S., Sumam, M.I.: Article: Named Entity Recognizer employing Multiclass Support Vector Machines for the Development of Question Answering System. International Journal of Computer Applications, vol. 25(10), pp. 40-46. (2011)
5. Desmet, B., Véronique, H.: Dutch named entity recognition using ensemble classifiers. In: Proceedings of 20th Meeting of Computational Linguistics in the Netherlands (CLIN), Landelijke Onderzoeksschool Taalwetenschap (LOT), (2010)
6. Ekbal, A., Bandyopadhyay, S.: A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi. Linguistic Issues in Language Technology, vol. 2(1), pp. 1-44, (2009)
7. Federico, M., Bertoldi, N., Sandrini V.: Bootstrapping named entity recognition for Italian broadcast news. In: Proceedings of ACL-02 conference on Empirical methods in natural language processing, vol.10, pp. 96-303. (2002)
8. Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named entity recognition through classifier combination. In: Proceedings of seventh conference on Natural language learning at HLT-NAACL 2003 Association for Computational Linguistics, pp. 168-171. (2003)
9. Kaur, D., Gupta, V.: A survey of named entity recognition in English and other Indian languages. In: Proceedings of the IJCSI, pp. 239-245. (2010)
10. Kudo, T.: CRF++, an open source toolkit for CRF. <http://crfpp.sourceforge.net>, 2005.
11. Kumar, S., Jha, G.N., Lalitha Devi, S.: Challenges in Developing Named Entity Recognition System for Sanskrit, In: Proceedings of Workshop on Indian Language and Data: Resources and Evaluation Workshop Programme, pp. 70-75. (2012)
12. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields for segmenting and labelling sequence data. In: Proceedings of ICML-01, pp. 282-289. (2001)
13. Lalitha Devi, S., Vijay, S.R.: Noun Phrase Chunker for Tamil. In: Proceedings of Symposium on Modeling and Shallow Parsing of Indian Languages, Indian Institute of Technology, Mumbai, pp. 194-198. (2006)
14. Malarkodi, C.S., Patabhi, R.K., Lalitha Devi, S.: Tamil NER–Coping with Real Time Challenges, In: Proceedings of Workshop on Machine Translation and Parsing in Indian Languages(MTPIL-2012), COLING, pp. 23-38 (2012)
15. Malarkodi, C.S., Lalitha Devi, S.: A Deeper Look into Features for NE Resolution in Indian Languages. In: Proceedings of workshop on Indian Language Data: Resources and Evaluation, LREC, Istanbul, (2012)
16. Minkov, E., Richard, C.W., William, W.C.: Extracting personal names from email: applying named entity recognition to informal text. In: Proceedings of conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 443-450. (2005)
17. Nadeau, D., Sekine S.: A survey of named entity recognition and classification, Linguisticae Investigationes, vol. 30(1), pp.3-26. (2007)
18. Patil, A., Sachin, P., Girish K.P.: Named Entity Extraction Using Information Distance. In: Proceedings of Sixth International Joint Conference on Natural Language Processing, pp. 1264-1270. (2013)

19. Rössler, M.: Corpus-based Learning of Lexical Resources for German Named Entity Recognition, In: Proceedings of Fourth International Conference on Language Resources and Evaluation (LREC'04), (2004)
20. Saha, S.K., Chatterji, S., Dandapat, S., Sarkar, S. Mitra P.: A hybrid approach for named entity recognition in Indian languages. In: Proceedings of IJCNLP-08 Workshop on NER for South and South East Asian Languages, pp. 17-24 (2008)
21. Vidanapathirana, N.P.: Agricultural information systems and their applications for development of agriculture and rural community a review study. In: Proceedings of 35th Information Systems Research Seminar in Scandinavia, (2012)
22. Vijayakrishna, R. Lalitha Devi, S.: Domain focused Named Entity for Tamil using Conditional Random Fields, In: proceedings of IJNLP-08 workshop on NER for South and South East Asian Languages, Hyderabad, India, pp. 59-66. (2008)
23. Wallach, H.M.: Conditional random fields: An introduction. Technical Reports (CIS), MSCIS-04-21, (2004)
24. Witten, I. H.: Text mining, Practical handbook of Internet computing. vol. 4(1), pp. 56-59. (2005)

Question Answering Based on Temporal Inference

Zeineb Neji ^{1*}, Marieme Ellouze ², Lamia Hadrich Belguith³

Faculty of Economics and management, Tunisia, Sfax
Computer department, Miracl laboratory, University of Sfax

zeineb.neji@gmail.com, mariem.ellouze@planet.tn,
l.belguith@fsegs.rnu.tn

Abstract. Inference approaches in Arabic question answering are in their first steps if we compare them with other languages. Evidently, any user is interested in obtaining a specific and precise answer to a specific question. Therefore, the challenge of developing a system capable of obtaining a relevant and concise answer is obviously of great benefit. This paper deals with answering questions about temporal information involving several forms of inference. We have implemented this approach in a question answering system called IQAS: Inference Question Answering System for handling temporal inference.

Keywords: Question answering system; temporal inference; several forms of inference.

1 Introduction

Advances in Natural Language Processing (NLP), Information Retrieval techniques (IR) and Information Extraction (IE), have given Question/answering systems (QA) a strong boost. QA have started incorporating NLP techniques to parse natural language documents, extract entities, resolve anaphora, and other language ambiguities [1]. In order to develop question answering capabilities, we believe that a large corpus of questions and answers that are based on temporal information should be discovered. In this paper, we focus on the task of question answering in Arabic by thinking of an approach which can improve the performance of traditional Arabic question answering systems for handling temporal inference. Obviously, any user is interested in obtaining a specific and precise answer to a specific question [2]. Therefore, the challenge of developing a system capable of obtaining a relevant and concise answer is obviously of great benefit. The challenge becomes huge when we try to automatically process a complex natural language such Arabic. This complexity is mainly due to the inflectional nature of Arabic. The situation gets worse, when we talk about the considerable lack of resources in general. Moreover, in our chosen field, research on temporal entity extraction in English, German, French, or Spanish, uses local grammars, and neural networks to detect temporal entities. These techniques do not work well directly for Arabic due mainly to the rich morphology and high ambiguity rate of Arabic.

In this paper, we propose a new approach dealing with the recognition and processing of temporal information for Question Answering (QA).

The remaining of this paper is organized as follow. In the next section, we give a short overview of QA systems with a special attention to the QA systems based on complex questions. After that, we describe our proposed approach and its different steps. Finally, we conclude this work and make suggestions for future researches.

2 Related Works

In this section, we present the earlier works related to question answering in Arabic. Despite extensive research in Arabic, the criteria represent a challenge to the automatic language processing systems [3]. In the last decade, the volume of Arabic textual data has started growing on the Web. Question-Answering systems represent a good solution for textual information retrieval and knowledge sharing and discovery. This is reason why a large number of QA systems has been developed and extensively studied recently. English, in particular, is very well served due to the fact that the majority of documents available on the internet are in English [4]. The question/answering systems in Arabic still few compared to the number of those developed in English, for instance. This is mainly due to the lack of accessibility to linguistic resources and the inflectional nature of Arabic.

Following a comprehensive survey of available Arabic QA systems, we realized that there are some well-known similar systems.

In fact, the technology of Arabic question answering has been studied since the 1970s starting with the AQAS system presented in [5]. This is the first system for Arabic. It is a knowledge-based QA system that extracts answers only from structured data.

ArabiQA is an Arabic question answering system dealing with factoid questions, integrates NER (Named Entity Recognition) module and adapts JIRS (Java Information Retrieval System) to extract passages from Arabic texts [6].

A deep study of sentence formulation and structures in Arabic interrogative continues with AQuASys [7] which is a Question Answering System designed to make it possible for the user to type in a question formulated in Arabic natural language, and designed to answer questions related to a named entity that can be of any type: person, location, organization, time, quantity, etc. Therefore, the system takes, as input, questions starting with interrogative words (من/who, ما/what, أين/where, متى/when, كم العددية/how many, كم الكمية/how much).

QArabPro [8], is an Arabic QA system that uses IR and NLP techniques to extract answers. It supports the factoid but excludes how and why.

Those approaches deal only with non-complex questions where the answers are selected from their corresponding short and simple texts. The challenge becomes greater when we try to create capabilities of processing complex questions and finding their answers from a collection of texts. An important component of this effort deals with the recognition and processing of temporal information for Question Answering (QA).

When asking a question that refers directly or indirectly to a temporal expression, the answer is expected to validate the temporal constraints. To achieve such functionality, QA systems need first, to deal with relations between temporal expressions and events mentioned in the question and, second, to rely on temporal inference to justify the answer. Whenever the answer to a question needs to be justified, if temporal expressions are involved, the justification must contain some form of temporal inference [9]. For example, the expected answer type of question Q1 is a Date:

Q1: متى تقلد المنصف المرزوقي رئاسة الجمهورية التونسية؟

Q1: When Moncef Marzouki has held the presidency of the Republic of Tunisia?

The expected answer type is an argument of the first event $Evt1$ =تقلد / held the presidency which has two more arguments: المنصف المرزوقي/Moncef Marzouki and الجمهورية التونسية / Republic of Tunisia.

The answer to Q1: 12 ديسمبر 2011 / 12 December 2011, extracted from the context:

P1: The doctor was elected as the interim President of Tunisia on 12 December 2011 by the Constituent Assembly of Tunisia with 153 votes for, 3 against, and 44 blank votes.

P1 : انتخب الدكتور رئيساً مؤقتاً لتونس في 12 ديسمبر 2011 بواسطة المجلس الوطني التأسيسي بعد حصوله على أغلبية 153 صوتاً مقابل ثلاثة أصوات معارضة وامتناع اثنين و 44 بطاقة بيضاء.

In the paragraph *P1*, the event $Evt2$ =انتخب/ elected which has as arguments الدكتور/ the doctor and تونس/ Tunisia. The event $Evt1$ differs from the event $Evt2$, but they are related. To justify the answer, these relations must be recognized into the temporal inference. In fact, it is through this example that we illustrate the importance of temporal inference to determine the full content.

In this paper, we present a Question Answering (QA) methodology to handle temporal inference by combining all these forms of inference.

3 The Importance of Temporal Information

In this section, we discuss some interesting applications that depend on temporal entity extraction to enrich our work. Time is an important dimension of any information space. However, in the last few years it has been studied in several areas.

- Temporal information is well-defined [10]: the relationship can be of the types before, overlap or after.
- Temporal information can be normalized [10]: every temporal expression referring to the same semantics can be normalized to the same value in some standard format.
- Temporal information can be organized hierarchically [10]: Temporal expressions can be of different granularities, e.g., of day type (December 20, 2015) or of year type (2015).

The temporal relationship between two events may allow answering a temporal query although no explicit temporal information is associated with the events [11].

Actually, processing questions that involve temporal inference relies on (1) the recognition of events/states and of entities that participate in them, (2) the relative ordering of events in the question and in the texts, (3) the temporal properties of the entities being questioned and (4) identification of the expected answer and its relations to temporal expressions mentioned in the question or candidate answers.

4 Proposed Approach

The proposed approach involves three main modules (Figure 1), namely : (1) question processing for interpreting the question, its temporal requirements and selecting candidate answers, (2) document processing, which includes indexation based on temporal information, finally (3) answer processing, where we start with the temporal inference before getting the answer.

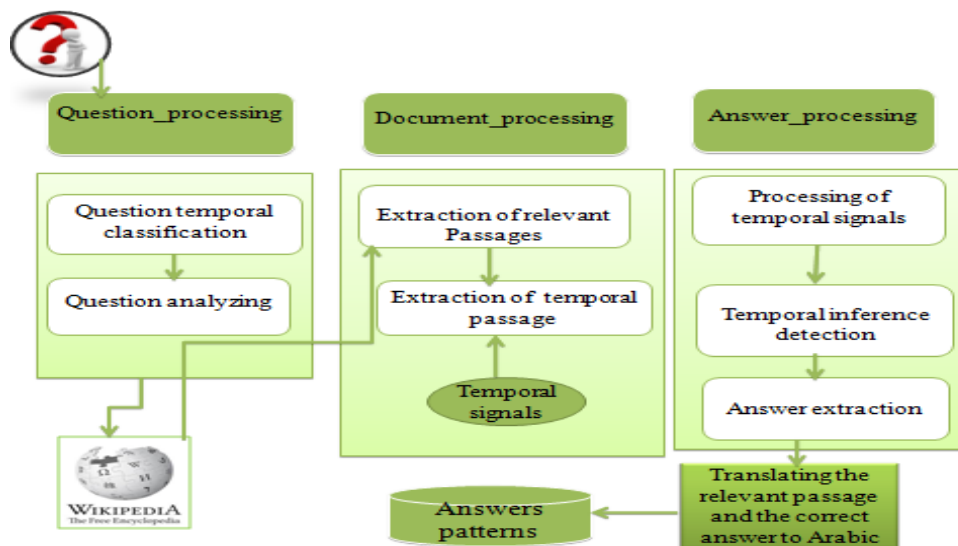


Fig. 1. Proposed approach

4.1 Question Processing

The objective of this process is to understand the asked question, for which analytical operations are performed for the representation and classification of the questions.

The first step of **question processing** is based on the classification of the questions referring to the temporal information extracted from the question. We have used the list of questions produced in TERQAS Workshop¹.

Some of the question classes are listed in Figure 2. This classification is based on the ways questions signal their time or event dependence and on how straightforward it is to determine the time at which information needs to be understood from a question so that will be possible to provide a suitable answer.

The classes of questions in Table 1, are characterized by the presence of a date and temporal signals, e.g. “since”, “after” and in some of them, we need to decompose the question to a temporal relation between events, indicated by a temporal signal.

Table 1. Classes of Temporal Questions

Question Class	Example
Time-Related	When was Moncef Marzouki president of Tunisia? How long did the Tunisian revolution last?
Event-Related	What must happen before the {Christmas} feast can begin in Poland?
Temporal-Order	Did John Sununu resign before or after George Bush’s ratings began to fall?
Entity-Related	How old was Mandela when he died?

We have experimented in a **first time** this classification using a set of 100 temporal questions and a set 100 associated answers extracted from temporal passages. The obtained results are very encouraging: **80%** of the temporal information selected from the suitable answer which contains the temporal information already expected from the question classification, **13%** to the unexpected answers (not correspond with the classification) and **7%** to the unfounded answers. In **second time**, we expand the number of questions; the results are shown in Table 2.

Some attempts were made to reach a better question analysis in the question answering task. Most of these attempts focused on keyword extraction from the user's question [13] made some query formulation and extracted the expected answer type, question focus, and important question keywords. To perform a better question analysis, the research of [14] analyzed questions by eliminating stop words, extracting named entities and classified the questions into Name, Date, Quantity, and Definition questions according to the question word used.

Table 2. Experiment results

Number of questions	Suitable answers
100	80
200	172
400	365

¹ TERQAS was an ARDA Workshop focusing on Temporal and Event Recognition for Question Answering Systems, www.cs.brandeis.edu/~jamesp/arda/time/readings.html

The research of [15] made some query formulation and extracted the expected answer type, question focus, and important question keywords. The question focus is the main noun phrase of the question that the user wants to ask about. For example, if the user's query is "What is the capital of Tunisia?" then the question focus is "Tunisia" and the keyword "capital" and the expected answer type is a named entity for a location. In our proposal, the step of analyzing the question is based on the elimination of stop words, extraction of the name entities and on the question classification.

4.2 Document processing

Extraction of the relevant document. QA systems benefit from keywords to quickly and easily find the relevant passages. All the documents are indexed with all these forms of information that enable the retrieval of the candidate's text passages.

Extraction of the relevant passage. Passages that do not contain time stamps or do not comply with the temporal relations that are searched are filtered out. Event recognition and classification as well as temporal expressions have been pointed out to be very important for our approach. TimeML [16] is a corpus annotated with: (a) time expressions; (b) events and (c) links between them. These annotations enable several forms of temporal inference [16] [17] [18]. The temporal information processing includes extracting events (TimeML EVENT tag), temporal expressions (TimeML TIMEX) and identifying temporal relations (TimeML TLINK tag).

The step of temporal passage retrieval has allowed only passages that contained at least one absolute or relative time expression. It also captures the event temporal orderings of the predicates and their relations to the answer structures.

4.3 Answer Processing

This module is responsible for selecting the response based on the relevant fragments of the documents. To be able to answer time-related questions, a question answering system has to know when specific events took place. For this purpose, temporal information can be associated with extracted facts from text documents [19].

Temporal inference. Either time expressions or events are related but are sometimes ambiguous. For example, the question Q: "how long did the Tunisian revolution last?" is classified to ask about a Time-Related, due to the presence of the question stem "how long".

`The Jasmine Revolution began on 18 December 2010 and led to the ousting of longtime president Zine El Abidine Ben Ali finally in 14 January 2011. Following the events, a state of emergency was declared.`

The answer that is inferred from this paragraph is "18 December 2010 –14 January 2011".

In Q, the event of the “Tunisian revolution” can be paraphrased by the “The Jasmine Revolution” expressed in the first sentence of the paragraph. The same mining is referred to the underlined expression “The events”.

The first reference (*The Jasmine Revolution began on 18 December 2010*) indicates a relation of **INITIATION** between the event «*The Jasmine Revolution*” and the fully specified temporal expression “*18 December 2010*” strengthened by the adverb “*be-gan*”. The second reference has an aspectual relation of **TERMINATION**, which is strengthened by the adverb “*finally*”.

The final inference enables the recognition of duration of an event when a time expression is identified for its initiation and for its termination. The correct answer is 27 days. The Automatic Translation provided by Wikipedia allows us to have by a simple and a quick projection to translate the pertinent passage and the right answer already found to Arabic. Such answers are important in Arabic QA system as they can be used to provide an answer from a document collection. We therefore decided to investigate the potential of those answers by acquiring patterns automatically

5 Answers patterns

It has been noted in several QA systems that certain types of answer are expressed using regular forms [20] [21]. For example, for temporal question like BIRTHDATEs (with questions like “When was X born?”/متى ولد فلان؟), typical answers are:

Mozart was born in 1756.	ولد موزارت في سنة 1756
Gandhi (1869-1948)	غاندي (1869 - 1948)
These examples suggest	These examples suggest
pattern like	pattern
<NAME> was born in <BIRTHDATE>	like
<NAME> <BIRTHDATE>-	<اسم> ولد في سنة <تاريخ ميلاد>
	<اسم><تاريخ ميلاد>-

When formulated as regular expressions, they can be used to locate the correct answer. Patterns are then automatically extracted from the returned documents and standardized to be then applied to find answers to new questions from a document collection. The precision of the patterns is calculated by cross-checking the patterns across various examples of the same type. This step will help to eliminate dubious patterns.

6 Conclusion

One of the most crucial problems in any Natural Language Processing (NLP) task is the representation of time. This includes applications such as Information Retrieval techniques (IR), Information Extraction (IE) and Question/answering systems (QA). This paper deals with temporal information involving several forms of inference in Arabic language. We introduced a methodology to compute temporal inference for QA that enables us to enhance the recognition of the exact answers to a variety of questions

about time. We have argued that answering questions about temporal information requires several different forms of inferences, including inferences that derive from relations between events and their arguments.

References

1. Vicedo, J., and Ferrandez, A. 2000. Importance of Pronominal Anaphora Resolution in Question Answering Systems. In Proceedings of the 38th Annual Conference of the Association for Computational Linguistics (ACL-00), 555–562.
2. Y. Benajiba, “Arabic Named Entity Recognition”, Ph.D. dissertation, Polytechnical University of Valencia, Spain, 2009.
3. A. M. Ezzeldin and M. Shaheen, “A survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends”, the 13th International Arab Conference on Information Technology ACIT’2012.
4. <http://www.internetworldstats.com/stats7.htm>
5. F. Mohammed, K. Nasser and H. Harb, “A knowledge based Arabic Question Answering system (AQAS)”, ACM SIGART Bulletin, pp. 21- 33, 1993.
6. Y. Benajiba, P. Rosso and A. Lyhyaoui, “Implementation of the ArabiQA Question Answering System’s components”, In: Proc. Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium. ICTIS-2007, Fez, Morocco, April 3-5, 2007.
7. S. Bekhti, A. Rehman, M. AL-Harbi and T. Saba, “AQUASYS: an arabic question-answering system based on extensive question analysis and answer relevance scoring”, In International Journal of Academic Research; Jul2011, Vol. 3 Issue 4, p45.
8. Hammo B., Abu-Salem H. and Lytinen S., “QARAB: A QA System to Support the Arabic Language”. Proceedings of the workshop on computational approaches to Semitic languages, pp. 55-65, Philadelphia, 2002.
9. S. Harabagiu, C. A. Bejan, “Question answering based on temporal inference.” in Proceedings of the AAAI-2005 Workshop on Inference for Textual Question Answering, 2005
10. Omar Alonso, J.Strötgen, A. Baeza, and M.Gertz: “Temporal Information Retrieval: Challenges and Opportunities”. TAW 2011: 1-8
11. S. Schockaert, D. Ahn, M. De Cock, and E. Kerre. “Question Answering with Imperfect Temporal Information”. In Proceedings of the 7th Conference on Flexible Query Answering Systems (FQAS 06), pages 647-658, 2006.
12. Ingria, B., and Pustejovsky, J. 2002. TimeML: “A Formal Specification Language for Events and Temporal Expressions”. <http://www.cs.brandeis.edu/~jamesp/arda/time/>
13. Brini, W., Ellouze, M., Mesfar, S., Belguith, L.H.: “An Arabic question-answering system for factoid questions”. In: IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009, pp. 1–7 (2009)
14. Rosso, P., Benajiba, Y., Lyhyaoui, A.: “Towards an Arabic question answering system”. In: Proc. 4th Conf. on Scientific Research Outlook Technology Development in the Arab world, SROIV, Damascus, Syria, pp. 11–14 (2006)
15. Harabagiu, S., Bejan, C., “An answer bank for temporal inference”, In Proceedings of LREC, 2006
16. Branimir Boguraev and Rie Kubota Ando. 2005. TimeML-Compliant Text Analysis for Temporal Reasoning. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005)

17. Sanda M. Harabagiu, Dan I. Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2001. The Role of Lexico-Semantic Feedback in Open-Domain Textual Question Answering. In Proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001), pages 274–281.
18. Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question Answering Based on Temporal Inference. In Proceedings of the AAAI-2005 Workshop on Inference for Textual Question Answering.
19. M. Pasca. Towards Temporal Web Search. In Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08), pages 1117–1121, 2008.
20. Lee, G.G., J. Seo, S. Lee, H. Jung, B-H. Cho, C.Lee, B-K. Kwak, J, Cha, D. Kim, J-H. An, H.Kim. 2001. SiteQ: Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP. Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, 437–446.
21. Wang, B., H. Xu, Z. Yang, Y. Liu, X. Cheng, D. Bu, and S. Bai. 2001. TREC-10 Experiments at CAS-ICT: Filtering, Web, and QA. Proceedings of the TREC-10

Statistical Approach for Spontaneous Arabic Speech Understanding Based on Stochastic Speech Recognition Module

Aymen Trigui, Naim Terbeh, Mohsen Maraoui, Mounir Zrigui

LaTice (Monastir team), Faculty of sciences of Monastir, Monastir, Tunisia
trigui.aymen@gmail.com, terbehnaim1987@gmail.com,
maraoui.mohsen@gmail.com, mounir.zrigui@fsm.rnu.tn

Abstract. This work is part of a big research project named "Oreillodule" aimed to develop tools for automatic speech recognition, translation, and synthesis for Arabic language. In this paper, our attention has mainly been focused on an attempt to present the semantic analyzer developed for the automatic comprehension of the standard spontaneous Arabic speech. We present a model of Arabic speech understanding system. In this model, both speech recognition module and semantic decoding module are based on statistical approach. In this work, we present and evaluate speech recognition module but we just explain the principle of Arabic speech understanding module.

Keywords: Speech understanding, Arabic language, Probabilistic model, semantic analyses, corpus.

1 Introduction

In the past 40 years there has been a significant research effort directed toward automatic speech recognition. Our work falls within the area of automatic understanding of the Arabic language, specifically in the context of finalized human / machine communication interfaces. The efficiency and performance of automatic spontaneous Arabic speech understanding system depend on its strength and its ability to overcome the difficulties of natural language processing among which some are linguistic and this concerns the understanding of written and spoken data. These problems are usually caused by the use of references, polysemic words, vague predicates, implicit form, etc. Others are due generally to the characteristics of spontaneous oral and in particular of the Arabic speech one.

The uses of statistical models for speech recognition and understanding have the advantage of being portable to other areas, or to multilingual applications [1]. In this work, we present a model for spontaneous Arabic speech understanding system. Both speech decoding and statement understanding are based on statistical approaches. We start in a first section by presenting the Arabic speech specificities and in the following section; we expose system architecture with a detail about each system components with its formal description.

2 Arabic Speech Specificities

In this section, we begin by listing the main characteristics of the Arabic speech and we detail its specificities.

2.1 Arabic Phoneme Set

Consonants. We can classify consonants according to several grammatical and phonetic criteria [2]: consonants articulated with vibration of the vocal cords and consonants that do not cause a vibration of the vocal cords, the crossing of air through the vocal tract gives rise to other varieties of sounds. The 28 Arabic consonants can also grammatically, be divided into two groups [3]:

- 14 solar consonants those are similar to the pronunciation of the « ل » (Al) in the « الشمس » word (the sun). With this consonants category we must pronounce the letter "Al" before the word.
- 14 lunar consonants those are similar to the pronunciation of the « ل » (Al) in the « القمر » word (the moon). With this consonants category we do not pronounce the letter "Al" at the beginning of the word.

Table 1. Consonants classification taking into account the transcription constraints

Lunar Consonants	أ ب ج ح خ ع غ ف ق ك ه م و ي
Solar Consonants	ت ث د ذ ر ز س ش ص ض ط ظ ل ن

Vowels. In Arabic language, we distinguish three short vowels (« ُ » (ضمة/dhamma:/), « ِ » (كسرة/kasra:/), « َ » (فتحة/fatha:/)) and three long vowels (the fatha « َ » extended by an alif "ا", the dhamma « ُ » extended by a waw "و" and kasra « ِ » is extended by a "ي"). The duration of a long vowel is about twice the size of a short one. These vowels are characterized by the vibration of the vocal cords. They are represented in the following table:

Table 2. Arabic language vowels classification

Short	َ / - /ِ - /ُ
Long	َ - ي - و

2.2 Other Vocalic Achievements

Semi-vowels. Arabic has two phonemes considered as semi-vowels: a bilabial spirant one « و » and a prepalatal spirant one « ي », called semi-vowels or sonants because of their kinship with the closed vowels « ُ » et « ِ ». These phonemes are used sometimes as consonants, sometimes as the corresponding vowel.

The “Soukoun” (السكون). The "Soukoun" (السكون) is not a vowel itself but it is the absence of vowel. Indeed, even if it is part of "Haraket" (الحركات) of Arabic it is not comparable to the other six vowels. The "Soukoun" is noted by a small circle « ْ ».

The “Tanwin” (التنوين). The three vowel diacritics may be doubled at the end of a word to indicate that the vowel is followed by the consonant n. These may or may not be considered harakāt, and are known as tanwīn (تَنْوِين), or nunation. The signs indicate, from right to left, -un, -in, -an.

In words containing “Tanwin” we listen the vocalic presence of the /N/ (Noon) phoneme but in reality when the word will be transcribed the (Noon) will not appear between the word’s consonants [4] [5]. We can cite as an example the “Tanwin” in the end of the word “مَكْتَبٌ” (the office).

The Geminat. A consonant may be doubled by wearing the /chadda:/ « ّ » (الشدة) signe, which represents an intensification in the pronunciation of a consonant. All consonants may be geminated and this gemination has a sense différenciative function. The geminated consonant is considered equivalent to two identical single consonants immediately follow each other. A gemination appears only in the position where a group of two consonants is allowed, in other words, between vowels [6][7]. The beginning and the end of each geminated consonant may be owned by different significant item, in other words a word or morpheme boundary can pass between them. These two consonants can also be separated in one term of a morphological couple. It is therefore right to consider the geminate consonants as groups of two similar consonants.

3 Understanding System Presentation

3.1 System Architecture

We present in Figure 1 the general architecture of the proposed Arabic speech understanding system.

Speech Recognition Module. This module is responsible to providing a text from an acoustic segment. All details concerning this component on the system are described in our previous work [6] and [7].

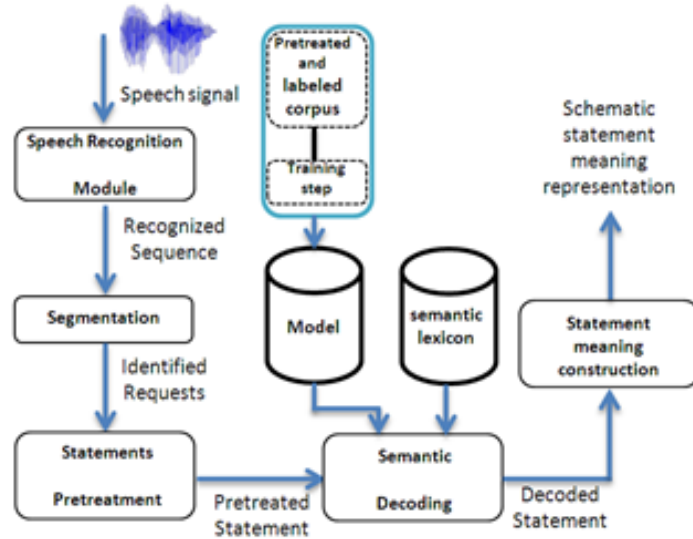


Fig. 1. The architecture of the proposed Arabic speech understanding system

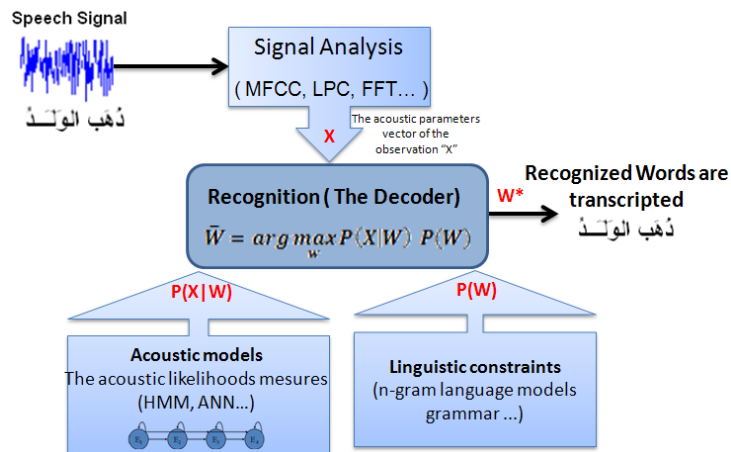


Fig. 2. General architecture of the speech recognition module

We present below the general architecture of the used Arabic speech recognition module.

From an acoustic segment the system generate a set of parameter vector (MFCC¹ in our case) the decoder will compare the result generated from the signal analysis module (MFCC) with different hypothesis using the both acoustic and linguistic modules; these too modules are previously trained using a tagged corpus. The module will provides the transcription corresponding to the most probable hypothesis.

¹ MFCC ; Mel Frequency Cepstral Coefficient

Segmentation. The role of this module is to segment statements transcribed by the speech recognition module. This treatment helps to identify the different requests of the speaker's message. The same message can consist of one or more requests at once. In this sense, it is necessary that the system can identify the different requests of the message, in order to interpret the user's request in its entirety.

Statements Pretreatment. The role of this module is to segment statements transcribed by the speech recognition module. This treatment helps to identify the different requests of the speaker's message. The same message can consist of one or more requests at once. In this sense, it is necessary that the system can identify the different requests of the message, in order to interpret the user's request in its entirety.

Semantics Decoding. It allows determining the meaning of each word in the statement.

Statement Meaning Construction. This module allows us to generate the entire attribute/value pairs.

Decoding pretreated semantic statements is based on a numerical model which encodes the rules of grammar, and on a semantic lexicon (see Figure 3 below). The semantic lexicon is a set of form associations: word/semantic features SEF describing the word meaning (see 1st definition)+ a set of syntactic feature SYF describing the word characteristics (gender, number and type). For example, the meaning of the word “الذاهب” (ranging) is described as follow:

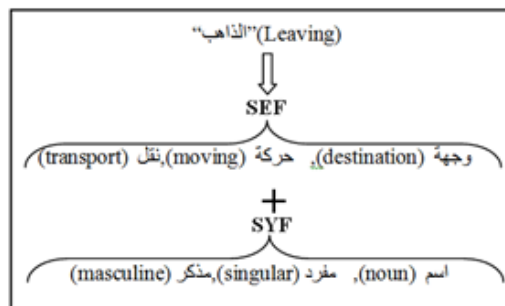


Fig. 3. Decoding pretreated semantic statements Using SEF and SYF

First definition: The semantic features set (SEF): An SEF is a set of three semantic features {D, C, and TM}, allowing the representation of the meaning of each non-empty word.

The first feature is concerning domain, referenced by “D”, it can specify the finalized treated application domain, for example it can provide information on touristic or Railways information etc.

The second feature is concerning semantic class, referenced by “C”, this feature specify the semantic class to which a word belongs to, This type of feature is used to

group synonymous words or words having the same semantic role in a specific application.

The third feature is a differential one and “TM” references him. It allows to oppose words to each other and to specify in what way they have a different meaning.

In their study [8], the authors create an automatic system of understanding of Arabic spontaneous speech. However, the proposed system is just for railway statements. Just one area. While in this work, we have tried to generalize their work. The authors presented the world with the triple (D, C, and TM), where "D" present the Domain (Area), and for them the domain is fixed (D = 1). "C" shows the semantic classes, this parameter is used to group sets of synonymous words or words that are semantically similar. For authors in these classes are created manually for the rail sector. "TM" allows words to oppose to each other and to clarify how they have different meanings.

Our idea is to:

- Determine the maximum possible existing domains (religious, sport, economics, computer science, politics, social sciences ...). In this way "D" will be an integer greater than 1. This allows us to cover the maximum possible of domain.
- "C" shows the semantic classes of each domain. Here, since it is impossible to determine the classes of all fields manually (as is done by the authors for the rail sector). So we used an automatic segmenter. It is an approach that allows to group synonymous words in one class. Where we get a set of semantic classes for each domain.
- "TM" will keep the same role as in [5].

4 Formal Description

Systems based on the language models attempt to determine the numerical score of a word sequence (statement) $S = m_1, m_2, \dots, m_i$, with the general formula is described by the equation (1) below:

$$P(S) = P(m_1) \cdot P(m_2 / m_1) \cdot \dots \cdot P(m_i / m_1, m_2, \dots, m_{i-1}) \quad (1)$$

In the case of the interpretation of a significant words sequence [12] M_1, M_2, \dots, M_n using a set of semantic features $SEF_1, SEF_2, \dots, SEF_n$ the model is trying to determine the score interpretation of each of these words, for each of these semantic features sets.

We denote by “I” the interpretation score that describe the meaning of the M_1, M_2, \dots, M_n words respectively with the semantic features sets $SEF_1, SEF_2, \dots, SEF_n$. I is measured as below:

$$\begin{aligned} P(I) &= P(SEF_1 \dots SEF_n / M_1 \dots M_n) \quad (2) \\ &= P(SEF_1 / M_1) \cdot P(SEF_2 / SEF_1, M_1 M_2) \cdot \dots \cdot P(SEF_n / SEF_1 \dots SEF_{n-1}, M_1 \dots M_{n-1}) \\ &= P(SEF_1 / M_1) \cdot P(SEF_2 / SEF_1, M_2) \cdot \dots \cdot P(SEF_n / SEF_1 \dots SEF_{n-1}, M_n) \end{aligned}$$

The transition from the first to the second line of the equation above is an approximation of the model, which considers that the probability of a SEF_i is conditionally dependent

only on the features sets of the current word M_i and not to those of the complete sequence. Fixing in advance the application domain, each significant words M_i can be interpreted through a set of semantic features using the form $SEF_i = (C_i, TM_i)$ and the above equation is transformed to equation 3 below:

$$P(I) \approx P((C_1, TM_1) / M_1) \times P((C_2, TM_2) / (C_1, TM_1), M_2) \times \dots \times P((C_n, TM_n) / (C_1, TM_1) \times \dots \times (C_{n-1}, TM_{n-1}), M_n) \tag{3}$$

To minimize the number of candidates features sets SEF and improve the performance of semantic decoder, we integrated, into the equation above, other sources of information (illocutionary nature and type of the statement) which can participates in the selection of SEF. Thus, the probability of interpreting each word M_i by a given features set $SEF_i = (C_i, TM_i)$, taking into account the nature and the type of statement noted by NT_j , is given by equation 4:

$$P(I) = P(SEF_1, \dots, SEF_n | NT_j, M_1 \dots M_n) = P(NT_j / M_1 \dots M_n) \times P((C_1, TM_1) / NT_j, M_1) \times P((C_2, TM_2) / NT_j, (C_1, TM_1), M_2) \times \dots \times P((C_n, TM_n) / NT_j, (C_1, TM_1), \dots (C_{n-1}, TM_{n-1}), M_n) \tag{4}$$

In the equation above $P(NT_j / M_1, M_2, \dots, M_{i-1}, M_i)$ is the probability that the statement is a NT_j type, knowing that the statement is made by the meaningful words M_1, \dots, M_n . Note that the first word is treated separately, by annotating it using the default class $C = request$.

The figure below shows an example of meaning construction using the designed model.

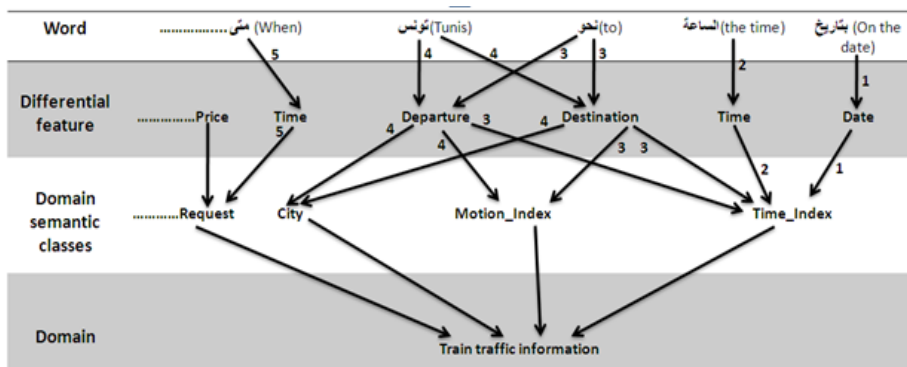


Fig. 4. Sample of statement meaning construction using our model

Figure 4 describes the structure and content of the semantic lexicon. The numbers used indicate the semantic features that can be used for the semantic representation of each word and each level of description.

5 Tests and Results

In order to evaluate our system we made an experiment using a corpus containing three domain; train reservation as the single domain used in [5] and we add to other domain; Book request in a library and touristic information as done in [9].

5.1 Corpus Collection

This corpus was collected by asking thirty-five different people to record vocal utterances relating to train reservation information request, touristic information request and book request. The following table provides information about the details about this task.

Table 3. Corpus collection details

Number of utterance	250
Number of speakers	35
Queries types	38
Data for Training	62%
Data for Test	38%

5.2 Evaluation

Some languages such as English, French, and German have platforms for evaluation understanding modules of dialogue systems. These platforms give to the community a large set of corpus of real annotated dialogues. However, this is not the case for the Arabic language where these resources are absent, with the exception of a few corpus distributed by ELDA/ELRA [10] [13]. The evaluation of corpus involves about 100 queries of different types (negation, affirmation, interrogation and acceptance), uttered spontaneously and manually transcribed. These requests correspond to scenarios dealing with information on the tourism fields. These scenarios are inspired from corpus MEDIA [11] and try to cover the input space The evaluation of the understanding module, with this evaluation corpus showed that this system generates 20 errors (average one error by 5 items). Measures of recall, precision and F-measure are respectively 72.00%, 69.00% and 75.69% and the average time to execute an utterance of 12 words is 0.279 seconds. Comparing these results with results obtained by other understanding modules [1], our system has provided fewer errors than many official sites such as UNISYS and MITRE.

6 Conclusion

We present in this paper a semantic decoder based on a hybrid language model, which allows integrating contextual, lexical, and semantic and illocutionary information at the same time. It allows, moreover, considering only the relevant sets of semantic features “SEF” in the history of the word to interpret. For this, we have developed a method to

automatically extract the relevant SEF which describe the meaning of words with semantic influence on the word to interpret. This is achieved, based on the concept of average mutual information brought by Rosenfeld (Rosenfeld, R. 1996). We intend eventually to evaluate our model by comparing it with other deployed models as models obtained by linear combination of language models well known as the maximum entropy.

References

1. Minker, W. (1999). *Compréhension automatique de la parole spontanée*. Editions Le Har-mattan.
2. Saidane, T., Zrigui, M., & Ahmed, M. B. (2005, May). Arabic speech synthesis using a concatenation of polyphones: the results. In *Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 406-411). Springer Berlin Heidelberg.
3. Mallat, S., Zouaghi, A., Hkiri, E., & Zrigui, M. (2013). Method of lexical enrichment in information retrieval system in Arabic. *International Journal of Information Retrieval Research (IJIRR)*, 3(4), 35-51.
4. Maraoui, M., Antoniadis, G., & Zrigui, M. (2009, July). SALA: Call System for Arabic Based on NLP Tools. In *IC-AI* (pp. 168-172).
5. Charhad, M., Zrigui, M., & Quénot, G. (2005, March). Une approche conceptuelle pour la modélisation et la structuration sémantique des documents vidéos. In *SETIT-3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, TUNISIA*.
6. Trigui, A., Maraoui, M., & Zrigui, M. (2010). Acoustic Study of the Geminant Effect in Standard Arabic Speech. *IPCV, 2010*, 192-196.
7. Trigui, A., Maraoui, M., & Zrigui, M. (2010, June). The gemination effect on consonant and vowel duration in standard Arabic speech. In *Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010 11th ACIS International Conference on* (pp. 102-105). IEEE.
8. Zouaghi, A., Zrigui, M., & Antoniadis, G. (2008). Compréhension automatique de la parole arabe spontanée. *Une Modélisation Numérique, Traitement Automatique des Langues (TAL 2008)*, 49(1), 141-166.
9. Lhioui, C., Zouaghi, A., & Zrigui, M. (2013). A combined method based on stochastic and linguistic paradigm for the understanding of arabic spontaneous utterances. In *Computational Linguistics and Intelligent Text Processing* (pp. 549-558). Springer Berlin Heidelberg.
10. Bahou, Y., Belguith, H. L., & Ben Hamadou, A. (2008). Towards a human-machine spoken dialogue in Arabic. In 6th Language Resources and Evaluation Conference (LREC 2008), Workshop HLT Within the Arabic World. Arabic Language and Local Languages Processing Status Updates and Prospects, Marrakech, Morocco.
11. Bonneau-Maynard, H., Rosset, S., Ayache, C., Kuhn, A., & Mostefa, D. (2005). Semantic annotation of the french media dialog corpus. In *Ninth European Conference on Speech Communication and Technology*.
12. Zouaghi, A., Zrigui, M., & Ben Ahmed, M. (2005). Un étiqueteur sémantique des énoncés en langue arabe. In Actes de la 12ème Conférence sur le Traitement Automatique des Langues Naturelles (TALNRECITAL 2005), Dourdan, France.
13. Villaneau, J., Ridoux, O., & Antoine, J. Y. (2004). LOGUS: compréhension de l'oral spontané. *Revue d'Intelligence Artificielle (RIA)*, 18(5-6), 709.

Initial Experiments on Russian to Kazakh SMT

Bagdat Myrzakhmetov and Aibek Makazhanov

National Laboratory Astana
Astana, Kazakhstan

{bagdat.myrzakhmetov, aibek.makazhanov}@nu.edu.kz

Abstract. We present our initial experiments on Russian to Kazakh phrase-based statistical machine translation. Following a common approach to SMT between morphologically rich languages, we employ morphological processing techniques. Namely, for our initial experiments, we perform source-side lemmatization. Given a rather humble-sized parallel corpus at hand, we also put some effort in data cleaning and investigate the impact of data quality vs. quantity trade off on the overall performance. Although our experiments mostly focus on source side pre-processing we achieve a substantial, statistically significant improvement over the baseline that operates on raw, unprocessed data.

1 Introduction

Machine translation from Russian to Kazakh poses certain challenges of both linguistic and technical nature. First of all, both languages belong to a group of so called morphologically rich or complex languages that have sophisticated inflection systems, and notoriously raise data sparseness and long range dependencies issues. This fact practically demands the data sparseness problem to be addressed. At the same time, however, both, source and target, languages being morphologically complex hinders the use of a common remedy – morphological segmentation.

When dealing with morphologically rich languages, including Kazakh [1, 2], most of the previous work performed morphological segmentation of words into sub-lexical units, such as isolated morphemes [1, 3], groups of morphemes [4] (both free and bound), or morpheme-like units obtained by automatic segmentation [5]. The motivation behind this approach is to reduce data sparseness by splitting off and/or removing some of the units, and hopefully improve alignment quality, as certain units in one language may correspond to words in another, e.g. case affixes often correspond to adpositions [1, 4]. The choice of a particular segmentation scheme, i.e. the decision of which sub-lexical units to isolate, remove, or group, is usually guided by the nature of translation languages in question, and based on the results of empirical comparison¹. When only one of the translation languages is morphologically complex and the other is not (e.g. mostly-analytic language such as Mandarin Chinese or English) candidate

¹ A notable exception is the work by Mermer [5] who proposed a language independent method that does not require any linguistic knowledge, and is capable of automatically choosing an optimal segmentation scheme by training a generative segmentation-translation model, which maximizes posterior probability of the training corpus.

Table 1. Realization of grammatical categories for Kazakh, Russian, and English: M - Morphological; A - Analytic; L - Lexical

#	Grammatical category	Kazakh	Russian	English
1	Number	M	M/L	M/L
2	Possession	M	A	A
3	Case	M/A	M/A	A/L
4	Gender	–	M/L	L
5	Agreement	M	M	M/L
6	Voice	M	M/A	A
7	Tense	M/A	M/A	M/A/L
8	Mood	M/A	M/A	A
9	Aspect	A+L	L	M/A/L
10	Negation	M/A	A	A

segmentation schemes can be devised intuitively (but not easily), provided sufficient familiarity with the grammar. However, when both languages are morphologically rich, and on top of it, typologically distant, devising such schemes may prove to be more complicated. This is exactly the problem with Russian and Kazakh.

While both, Russian and Kazakh, are morphologically complex, they employ different inflection strategies. Russian, a Slavic language, is usually classified as fusional, i.e. a language that “packs” several grammatical categories into a single affix. In contrast Kazakh, as a Turkic language, employs agglutination and “stacks” grammatical categories one per affix². While corresponding many-to-one mapping of affixes through common grammatical categories is possible, it has to account for idiosyncrasies of grammar of both languages, e.g. Russian numerals agree with nouns in case, and adjectives – in case, number and gender, while neither type of agreement is found in Kazakh³. Even if such a sensitive segmentation scheme could be designed, the issue does not end there: richness of morphology does not, of course, imply a language’s complete reliance on it. Many things in both languages can be also (or exclusively) expressed analytically and/or lexically. A very general, and by no means authoritative, summary of surface realization of major grammatical categories for both languages (and English as an example of morphologically “poor” language) is given in Table 1. Notice that only verbal agreement is realized by the same means, namely morphologically, and for other nine categories means of grammeme encoding allow certain variation. Thus, from a linguistic perspective, for the given language pair we are faced with sparseness on both sides and difficulty in designing morphological segmentation schemes.

From technical perspective, there is another challenge that concerns mostly Kazakh in its lack of resources for our particular purposes. By and large the language is being actively studied, and there exist monolingual corpora [6, 7], and ongoing research on

² For example, consider plural genitive of “language”: (*rus*) *jazyk-ov* – language-PL.GEN; (*kaz*) *til-der-ding* – language-PL-GEN. Notice how number and case are encoded in a single suffix in Russian (*rus*), and in two successive suffixes in Kazakh (*kaz*).

³ Thus if we were to segment Russian numerals and adjectives and split off corresponding suffixes their alignment to anything except null would be incorrect.

morphological processing [8–13] and syntactic parsing [14–16]. However, except for a rather small and noisy OPUS corpus [17] there are no Russian-Kazakh parallel corpora⁴ and the only tool for automatic morphological disambiguation of Kazakh available to us⁵ was reported to have accuracy of 86%, which we considered to be low enough to question the results of experiments with segmentation: would possible misalignments be shortcomings of a chosen segmentation scheme or results of incorrect morphological analysis and disambiguation. Thus, due to the lack of resources to process the target side and no solid approach to the design of suitable segmentation scheme for the particular pair of languages, in our first attempt to tackle the problem we settle for discussing potential challenges and conducting straightforward experiments.

To our knowledge, the present work is the first to address the problem of Russian to Kazakh SMT. Our contribution is a rather modest one: for the initial experiments, we are testing waters by performing a source-side lemmatization and dictionary-based data cleaning. The aim is to see what improvement one can get from a basic, light-weight morphological pre-processing, such as lemmatization, and to quantify data quality vs. quantity trade-off. The intuition behind lemmatization was to reduce sparseness by performing a “poor man’s segmentation”, where all the inflections of the source side morphologically complex language get trimmed off. The data cleaning step was necessitated by the fact that bulk of the data we used was misaligned and noisy. The results we obtained show that just source-side pre-processing gives a net improvement of 0.98 BLEU points (6.3% relative), and, when coupled with a data cleaning procedure, the approach results in 1.49 net and 9.5% relative improvement over the baseline.

The rest of the paper is organized as follows. In Section 2 we discuss the related work. In Section 3 we proceed to the description of the parallel corpus that we used for training, testing and tuning our models and the process of its collection and alignment. In Section 4 we describe our experiments and report results. Lastly, we provide concluding remarks and discuss future works in Section 5.

2 Related Work

There is a number of works on rule-based machine translation involving Kazakh language with the main focus on building bilingual dictionaries and structural transfer rules, and developing word sense disambiguation techniques for the open source Aperi-tium system [19] to translate to/from Russian [20] and English [21, 22].

As for the statistical machine translation research involving Kazakh, two studies concerned with Kazakh to English SMT [1, 2] perform morphological segmentation on the source side, using respectively Morphessor [23] and the HFST-based [24] Kazakh FST [11]. Both studies report relative improvement over the word-based baselines. Assylbekov and Nurkas [1] devise several segmentation schemes, and evaluate each of them. Their results suggest that removing 3rd person possession and genitive, accusative case markers from both nominals and non finite verbs, and doing the same for

⁴ At the time of experiments a Russian-Kazakh parallel corpus of approximately 342K sentence pairs described by Assylbekov et al. [18] was not available to us.

⁵ At the time of experiments a disambiguation tool with reported accuracy of almost 91% developed by Assylbekov et al. [8] was not available.

agreement markers, while splitting off much of the rest of the inflections, yields the best results. Interestingly enough, when faced with the problem of the absence of an accurate morphological disambiguation tool for Kazakh, the authors chose to reduce (not to resolve completely) ambiguity by using a constraint grammar-based tool, and to replace remaining ambiguous analyses by the first analysis returned by this tool. Unfortunately, the authors did not measure or in any way analyze the effect of incorrect disambiguation on the quality of produced alignments and translation.

Assylbekov et al. [18] report on building a Russian-Kazakh parallel corpus of around 342K sentence pairs, which, unfortunately, was not available at the time when we conducted our experiments. The authors describe various data pre- and post-processing techniques that improve the quality of sentence alignment. Among other methods the authors describe a dictionary-based re-alignment of lemmatized sentence pairs, a technique that we use in our data cleaning step. According to the authors this technique increases the portion of correctly aligned sentences (as measured by an automatic learner-based estimator) in a given bitext by 2% compared to the standard length-based alignment procedure.

Various segmentation schemes are also considered in works dealing with translation from and to Russian. Lo et al. [25] employ source-side lemmatization for the Russian-to-English translation task. However, unlike what we do in the present work, the authors use lemmatization only at the word alignment step and restore original surface forms before estimating the remaining parameters of the model. The intuition is to obtain accurate word alignments through lemmatization, while being able to use valuable information, such as case and agreement markers, encoded in inflected forms. For the same translation task Borisov and Galinskaya [26] propose a sophisticated segmentation scheme, consisting of a number of rules, that cover nouns, adjectives, various verb forms, and a catch-all rule that covers all possible remaining cases, except ambiguous analyses, which are simply skipped. Apart from “usual suspects” such as case and agreement markers, the authors consider isolation of comparison degree affixes from adjectives (for some reason living out this category for adverbs) to account for irregular forms in English. Remaining inflections for this parts of speech, i.e. gender, number, and case, are removed as they are not defined for English adjectives. The authors report a small improvement over an already strong baseline, and respective reduction of 35% and 29% (best case) in word types and OOV rate for Russian.

3 Data Set

Our data set consists of a portion of the OPUS corpus [17], on-line news (<https://ortcom.kz>), a collection of historical essays (<https://e-history.kz>), and legal texts (<https://akorda.kz>). When obtained the Russian-Kazakh portion of the OPUS corpus contained 92035 parallel sentences that came mostly from software documentation and movie sub-titles. However, upon visual inspection we have noticed some repetitions, corrupted text bits, and clearly misaligned pairs. After manually removing such instances, we ended up with more than 56K sentences. News, essay, and legal data were aligned using Hunalign [27]. The aligned data were manually checked and around 2.5K sentence pairs were held out for testing and tuning, while remaining

Table 2. Quantitative description of the data set

	# sentences	# tokens	# unique tokens
Kazakh			
Training	69889	447499	68547
Testing	1004	15563	5653
Tuning	1510	21691	7065
Russian			
Training	69889	475388	68884
Testing	1004	16509	5958
Tuning	1510	22936	7251

Table 3. Domain distribution on the data set

Domain	Training		Testing		Tuning	
	# sen-s (%)	# tok-s (%)	# sen-s (%)	# tok-s (%)	# sen-s (%)	# tok-s (%)
OPUS	56695 (81.5)	425431 (46.1)	-	-	-	-
News	10027 (14.3)	383079 (41.5)	491 (48.9)	14766 (46.0)	1260 (83.3)	37606 (84.3)
Essay	1402 (2.0)	49963 (5.4)	72 (7.2)	2241 (6.9)	-	-
Law	1765 (2.2)	64414 (7.0)	441 (43.9)	15065 (47.1)	252 (16.7)	7071 (15.7)

13K+ pairs together with the cleaned OPUS data comprised the training set. Table 2 shows the counts of sentence pairs, running and unique tokens contained in the data set. The distribution of the data across domains is shown in Table 3.

4 Experiments and Evaluation

In our experiments we build and compare two models: (i) the baseline model that assumes no pre-processing of the input, and (ii) the source lemmatized (SL) model which is run on the data whose source side (Russian) has been lemmatized. For lemmatization we use Mystem [28], a freely available stemmer for Russian. We proceed to evaluate both models in two settings: on complete and cleaned training sets respectively. For cleaning we again use hunalign [27], but this time we also use a Russian-Kazakh dictionary⁶ with about 116K entries. The tool scores existing alignments, and we remove those that ranked among the bottom 10%. Because dictionary entries are lemmatized, during cleaning we perform lemmatization on both source and target sides of the training set, and later restore the target side of the cleaned data. For target side lemmatization we use a data-driven morphological disambiguator for Kazakh [10].

We implement the models using the Moses toolkit [29], setting the distortion limit parameter to -1 (infinity) to account for long range dependencies and free word order of the languages. Remaining parameters are estimated on the tuning set with the help of the MERT [30] procedure. We train a 3-gram language model smoothed with the modified

⁶ <http://mtdi.kz/til-bilimi/sozdikter/oryssha-kazaksha>

Table 4. Main results

Model	Training set	
	Complete	Cleaned
Baseline	15.62±0.08	16.02±0.11
Source lemmatized	16.60±0.08	17.11±0.09
Statistical significance, %	98.6	100.0

Kneser-Ney [31] algorithm on the target side of the training data and a portion of the Kazakh Language Corpus [6] that contains slightly more than 1.3M running tokens. The models are evaluated in terms of BLEU [32] metric. The results are averaged over three independent tuning runs and reported in Table 4 together with the standard deviations and results of the statistical significance tests. Statistical significance is calculated using bootstrap resampling technique [33] for 1000 samples under the null-hypothesis that the SL model outperforms the baseline.

As it can be seen on the complete data set the SL model scores at 16.6 points against 15.6 points of the baseline, achieving 6.3% relative improvement. Data cleaning boosts the performance of both models on approximately half a point. On the cleaned data the SL model outperforms the baseline again, and achieves relative improvement of 6.8%. Thus, after cleaning and lemmatization, we improve 9.5% over the baseline that operates on the raw, unprocessed data.

Lastly, in terms of combating data sparseness, let us note that after lemmatization, the number of unique tokens on the source side has been reduced to 62.5% (from 68884 to 25824), and the size of the phrase table has been reduced to 12.3% (from 1070916 to 939087 entries).

5 Conclusion and Future Work

We have conducted initial experiments with Russian to Kazakh SMT. Our findings suggest that even light weight morphological processing, such as lemmatization on the source side, provides substantial improvement over the word-based baseline that assumes no pre-processing of the input. We have also showed that noise reduction in the training set can be beneficial as well, although the improvement is a less drastic one.

For the future work we plan to closely investigate various strategies of morphological segmentation for these languages. We also plan to enrich our data with grammatical annotation in order to experiment with factored models [34].

Acknowledgments

This work has been funded by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan under the targeted program O.0743 (0115PK02473) and by the Nazarbayev University under the research grant 064-2016/013-2016.

References

1. Assylbekov, Z., Nurkas, A.: Initial explorations in Kazakh to english statistical machine translation. In: The First Italian Conference on Computational Linguistics, CLiC-it. (2014)
2. Kartbayev, A.: Learning word alignment models for kazakh-english machine translation. In: Integrated Uncertainty in Knowledge Modelling and Decision Making - 4th International Symposium, IUKM 2015. (2015) 326–335
3. Bisazza, A., Federico, M.: Morphological pre-processing for turkish to english statistical machine translation. In: IWSLT 2009. (2009)
4. Oflazer, K., El-Kahlout, I.D.: Exploring different representational units in english-to-turkish statistical machine translation. In: Proceedings of the Second Workshop on Statistical Machine Translation, Association for Computational Linguistics (2007) 25–32
5. Mermer, C.: Unsupervised search for the optimal segmentation for statistical machine translation. In: Proceedings of the ACL 2010 Student Research Workshop, Association for Computational Linguistics (2010) 31–36
6. Makhambetov, O., Makazhanov, A., Yessenbayev, Z., Matkarimov, B., Sabyrgaliyev, I., Sharafudinov, A.: Assembling the kazakh language corpus. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, Association for Computational Linguistics (October 2013) 1022–1031
7. Altenbek, G., Xiao-long, W.: Kazakh segmentation system of inflectional affixes. In: Joint Conference on Chinese Language Processing, CIPS-SIGHAN (2010) 183–190
8. Assylbekov, Z., Washington, J., Tyers, F., Nurkas, A., Sundetova, A., Karibayeva, A., Abduali, B., Amirova, D.: A free/open-source hybrid morphological disambiguation tool for Kazakh. In: TurCLing 2016. (2016) 18–26
9. Kessikbayeva, G., Cicekli, I.: A rule based morphological analyzer and a morphological disambiguator for kazakh language. *Linguistics and Literature Studies* 4(1) (2016) 96–104
10. Makhambetov, O., Makazhanov, A., Sabyrgaliyev, I., Yessenbayev, Z.: Data-driven morphological analysis and disambiguation for kazakh. In: Proceedings of the 2015 Computational Linguistics and Intelligent Text Processing, Part I, Cairo, Egypt, Springer International Publishing (2015) 151–163
11. Washington, J., Salimzyanov, I., Tyers, F.: Finite-state morphological transducers for three kypchak languages. In Chair, N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, European Language Resources Association (ELRA) (may 2014)
12. Makazhanov, A., Makhambetov, O., Sabyrgaliyev, I., Yessenbayev, Z.: Spelling correction for kazakh. In: Proceedings of the 2014 Computational Linguistics and Intelligent Text Processing, Kathmandu, Nepal, Springer Berlin Heidelberg (2014) 533–541
13. Makazhanov, A., Yessenbayev, Z., Sabyrgaliyev, I., Sharafudinov, A., Makhambetov, O.: On certain aspects of kazakh part-of-speech tagging. In: Application of Information and Communication Technologies (AICT), 2014 IEEE 8th International Conference on. (Oct 2014) 1–4
14. Altenbek, G., Wang, X., Haisha, G.: Identification of basic phrases for kazakh language using maximum entropy model. In: COLING. (2014) 1007–1014
15. Tyers, F.M., Washington, J.: Towards a free/open-source universal-dependency treebank for kazakh. In: 3rd International Conference on Turkic Languages Processing (TurkLang 2015), Kazan, Tatarstan (2015) 276–290
16. Makazhanov, A., Sultangazina, A., Makhambetov, O., Yessenbayev, Z.: Syntactic annotation of kazakh: Following the universal dependencies guidelines. a report. In: 3rd International Conference on Turkic Languages Processing (TurkLang 2015), Kazan, Tatarstan (2015) 338–350

17. Tiedemann, J.: Parallel data, tools and interfaces in opus. In: LREC. (2012)
18. Assylbekov, Z., Myrzakhetov, B., Makazhanov, A.: Experiments with Russian to Kazakh Sentence Alignment. *Izvestija KGTU im.I.Razzakova* **38**(2) (2016) 18–23
19. Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Sánchez-Martínez, F., Ramírez-Sánchez, G., Tyers, F.M.: Apertium: A free/open-source platform for rule-based machine translation. *Machine Translation* **25**(2) (June 2011) 127–144
20. Rakhimova, D.: Research of problem of the semantic analysis and sythesis of pretext in the russian-kazakh machine translation. In: 3rd International Conference on Turkic Languages Processing (TurkLang 2015), Kazan, Tatarstan (2015) 59–67
21. Assem Shormakova, Aida Sundetova, A.S.: Features of machine translation of different systemic languages using an apertium platform (with an example of english and kazakh languages). *JSCSE* (2013) 255–259
22. Sundetova, A., Forcada, M., Tyers, F.: A free/open-source machine translation system for english to kazakh. In: 3rd International Conference on Turkic Languages Processing (TurkLang 2015), Kazan, Tatarstan (2015) 78–91
23. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)* **4**(1) (2007) 3
24. Linden, K., Silfverberg, M., Axelson, E., Hardwick, S., Pirinen, T. In: HFST-Framework for Compiling and Applying Morphologies. Volume Vol. 100 of Communications in Computer and Information Science. (2011) 67–85
25. Lo, C.k., Cherry, C., Foster, G., Stewart, D., Islam, R., Kazantseva, A., Kuhn, R.: Nrc russian-english machine translation system for wmt 2016. In: Proceedings of the First Conference on Machine Translation, Association for Computational Linguistics (2016) 326–332
26. Borisov, A., Galinskaya, I.: Yandex school of data analysis russian-english machine translation system for wmt14. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, Association for Computational Linguistics (2014) 66–70
27. Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., Trón, V.: Parallel corpora for medium density languages. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4* **292** (2007) 247
28. Segalovich, I.: A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In: MLMTA. (2003)
29. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, Association for Computational Linguistics (2007) 177–180
30. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, Association for Computational Linguistics (2003) 160–167
31. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. Volume 1., IEEE (1995) 181–184
32. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics (2002) 311–318
33. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP 2004, Association for Computational Linguistics (2004) 388–395
34. Koehn, P., Hoang, H.: Factored translation models. In: EMNLP-CoNLL. (2007) 868–876

Contribution to the Achievement of a Spellchecker for Arabic

Khairredine Bacha, Mounir Zrigui

University of Tunis, High School of Sciences and Techniques of Tunis,
Laboratoire LaTICE, Tunisia
{khairi.bacha@gmail.com, mounir.zrigui@fsm.rnu.tn}

Abstract. The objective of this work is to perform a spell check tool that analyzes the text entered in search for possible misspellings. This tool will suggest possible corrections for each misspelled word in the text. This work will require the presence of a reference dictionary of words in the arabic language. These objective Were Accomplished with resources, effective methods, and approaches. First experimental results on real data are encouraging and provide evidence of the validity of the design choices. They also help to highlight the difficulty of the task, and suggest possible developments.

Keywords: Spellchecker, Arabic, dictionary, error detection.

1 Introduction

Natural Language Processing (NLP) is a discipline that closely associates linguists and informaticians. It is based on language, formalisms (representation of information and knowledge in formats interpretable by machines), and computer science. This is the set of methods and programs that allows computer processing of language data, but when this treatment takes into account the specificities of human language. There are language data processing (writes files, backups or other) that are not part of natural language processing [1]. Indeed, it is integrated with informatic tools used daily by millions of people worldwide. A spellchecker detects, in a given input text, the words that are incorrect. A spell checker detects the same time spelling errors and look, for the correct word most likely [2].

Construction of automatic text correction systems is one of the oldest applications of Natural Language processing techniques, since, according to Mitton [3], the first systems of automatic detection appeared in the late 50s. A spell checker performs two essential functions, one after another: first detecting, then correcting spelling errors. Methods for detection and correction work in three ways: First error detection orthographically consisting of foreign words in the language. Then the whole word of correction is to correct the word previously detected in the single recital regardless of the words that surround it. Finally, the detection and contextual error correction where each word is considered taking into account the context; Which corrects spelling mistakes even when it consists of words found in the language but are misplaced [4].

It is in this context that our research lies. For this, we will implement an orthographic equalizer that analyzes the text entered to find any spelling errors. This tool will suggest possible corrections for each misspelled word in the text. This work will require the use of reference dictionary of Arabic words [5].

2 State of the Art in Spelling Correction

Spell checking is to find the word (s) nearest incorrect words in a text of a language, based on similarity and distance inter words. Several researchers have studied the problem ; and through their efforts various techniques and many algorithms have been developed [6]. The first studies were devoted to determining the different types of elementary spelling errors, called publishing operations including: insertion, deletion, permutation, and substitution [7].

The main techniques used for the identification of erroneous words in a text are either based on the analysis of n-grams, or about searching a [8] dictionary. An algorithm for the detection based on a dictionary is given by Peterson [9]. Another modeling proposed by Pollock and Zamora [10] is carried out by comparing the alpha-codes with the erroneous word. It is to associate each word in the dictionary with its alpha -code, hence the need to have two dictionaries: one for the words and another for their alpha -codes. They propose to correct those words containing only one error, which form 90-95% of the errors. A particular problem concerns the errors that result in words that exist in the lexicon. According to Mitton [3], this applies to 16% of spelling errors. Oflazer proposed a new approach for the correction of a wrong word that consists in browsing the dictionary controller by calculating for each transition the cut -off edit distance without exceeding the threshold defined in [11] algorithm.

Despite the availability of a set of methods for spell checking, we find out that we do not yet have robust correction software that can handle appropriately all mistakes in the written text, seeing the bad scheduling solutions suggested when correcting.

2.1 Systems for Arabic

The major problem is that the Arabic language is very rich in morphology, and it has many exceptions. Moreover the lack of vowels and words of agglutination make treatment a more difficult task. The most successful achievements for the Arabic language are:

- **BenOthmaneZribi and Zribi (1999)** evoke the special problems to correct Arabic. Words must sometimes be vowelized. Moreover, Arabic is an agglutinative language which uses affixes and enclitics (pronouns) and proclitics (adverbs, prepositions and conjunctions). In addition, this language contains many lexemes that are very similar to each other. Candidate proposals for the correction of a word can be very numerous. The corrector is accompanied by a morphological analyzer, which cuts shape proclitic, radical and enclitic [12].
- **ShalanSpell-Checker (2003)**: a correction can detect and correct common spelling mistakes in Arabic based on the technique of N-gram [13].

- **Haddad-Yaseen Spell Checker (2007):** It is a hybrid model for spell checking and correction of Arabic words, based on the recognition of semi-isolated words [14].
- **Zerrouki-BallaSpell-Checker (2009)** developed a spell checker for the Arabic language based on N-gram [15].
- **Hasan Muaidi and Rashal al Tarawneh Spell-Checker (2012):** It is a simple and flexible spell checker for the Arabic language based on N-gram scores technique (matrix). The recognition rate of the proposed spellchecker reached 98.99% [16].
- **Gueddah, Yousfi and Belkasm (2012)** proposed a typical and effective variant of edit distance by integrating the frequency matrices editing errors in the Levenshtein algorithm in order to perfect the correction and scheduling error suggestions committed in the seizure of documents in Arabic [17].

2.2 Error Correction Techniques

Situations where we may use detection or automatic correction of spelling errors are very diverse: isolated words or context, common nouns, proper names, labels ... performance errors are not random but systematic such as the inclusion or omission of letters. They note that a list of proposed corrections must be as short as possible and the correct word must appear in such a high order as possible. The spelling correction process: it is looking for a word in a dictionary, and if the word is not there, the search for the most likely words to represent the correct spelling of the word [18]. Error usually single words of correction techniques can be divided into subcategories:

- **Distance of Levenshtein:** This is a mathematical distance measure that gives a similarity between two character strings. It is equal to the minimum number of characters you need to remove, insert or replace to move from one channel to another. It was proposed by Levenshtein in 1966. It measures the similarity between words by computing an edit distance. The edit distance is defined as the minimum number of elementary edit operations needed to transform a wrong word to a dictionary word. Thus, to correct a misspelled word, it retains a set of solutions that requires the least possible editions of operations. It is also known under the names of edit distance or time dynamic deformation, including pattern recognition, especially in voice recognition [19].
Called Levenshtein distance between two words M and P the minimum cost to go from M to P by performing the following basic operations:
 - Substitution of M character in a character P;
 - Added in M P a character;
 - Delete character Mr.
- **Correction rules:** Yannakoudakis and Fawthrop offer two spellers by rules. One is based on research of similar words in a dictionary, following some error rules; the second searches in the dictionary words that differ by one or two characters of the unknown channel and checks whether an error rule can be applied. [20] Emirkanian Bouchard and uses heuristics to correct spelling errors. The radical words is looked in a dictionary, which also contains information on the correct suffix and suffix frequent errors. A Dictionary of suffixes allows you to find suffixes invalid language [21].

- **The technique of n-grams:** N-grams are n groups of letters constituting a word substring. The most common are bigrams, consisting of two letters and three letters of the trigrams. In general, the techniques of n-grams examine each of the n-grams constituting the input string and looking its presence or its frequency of occurrence in a precompiled table containing statistics on the most frequent n-grams. For de Heer, the trigrams are the smallest units which, combined, are significant to the meaning of language. [22] Angell and his colleagues present a method based on common trigrams between the unknown word and dictionary words. The candidate words are found through a dictionary of trigrams which lists all words that contain the same trigram [23].
- **Probabilistic Technique:** The N-gram based technology naturally led in the probabilistic technique both text recognition and spelling correction paradigms. It requires a very large corpus of text in order to establish the table of n-grams. Research techniques in the dictionary only prospect if the input string appears or not in the list of valid words. If the string is missing from the dictionary ; then, it is called erroneous. The dictionary access time becomes prohibitive when the size of the latter exceeds a few thousand words. This problem was addressed in three distinct views through effective search algorithms [24]; via partitioning and organization of dictionaries [25], or via the techniques of editing distance [26] and morphological processing. The most exploited technique to gain access to dictionaries time is the technique of hash. Most existing spellers are semi automatic, assist the user by offering a set of candidates close to the erroneous word [27].

3 Spellchecker implementation

We will see in this part of resource development, implementation and evaluation of our spelling system. We first describe the principle and the overall architecture of our correction. Then we present the different techniques and spell checking tools. We will finish with an evaluation of our system.

3.1 Principle

Our spellcheck prototype must be able to detect misspelled words in a text input and suggest corrections. It addresses a set of operations for each misspelled word: Error detection is often done by considering one by one the words of the text to correct in isolation. Every word of the text is compared with dictionary words. Any word not found in the dictionary is considered wrong.

AL-Mohit Dictionary is a multifunctional electronic dictionary for Arabic. It studied the macrostructure and microstructure of electronic dictionaries of the Arabic language [5]. IT is very rich in grammatical information and meaning and definitions of words. This dictionary can be seen as a structure composed of linguistic objects. Among these objects we can find: the headword, pronunciation, grammatical categories that can have this headword (فعل, اسم, مصدر, صفة) definitions, translations, examples. This dictionary, implements the form of a relational database, contains in its first version fifteen entities or files left in two main branches: a verbal branch and nomina l branch. When an error is detected, correction selects a series of words likely

to be the correct version of the string to correct. These words are selected using various techniques.

Generation of possible corrections can be done using the concept of Levenshtein distance, consulting a predefined rule base and adding spaces in the middle of each misspelled word. The authorization of the correction to candidates chains considers the measure used in the selection step, as well as statistical measures (such as the frequency of occurrence of words, or the word most frequently chosen during preliminary meetings with the same error). Finally, interactive step allows the user to supervise the correction. It can adopt one of two attitudes: First, correct the erroneous word by selecting one of the candidates proposed by the checker. Then, modify the wrong word.

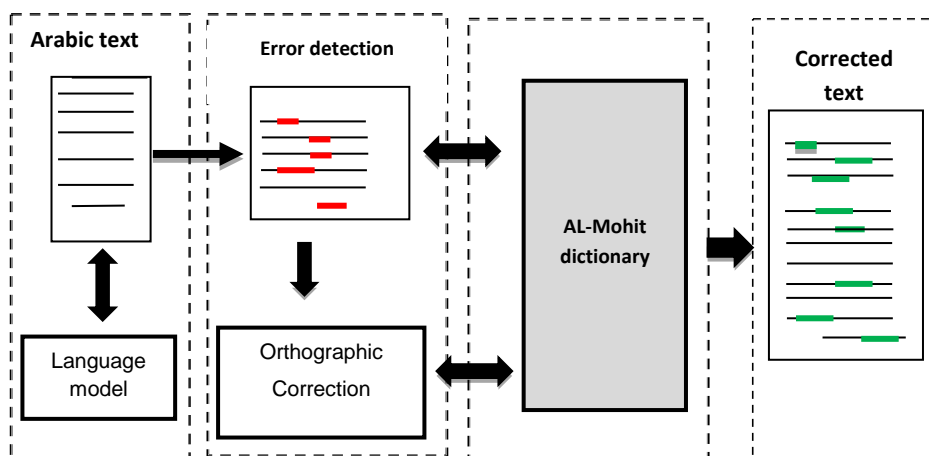


Fig. 1. General architecture of the prototype: spellchecker

3.2 Global architecture

In this section, we briefly present the resources that have been made to develop and test our first spellchecking prototype: It seems impossible to have a large corpus, which represents all possible co-occurrences and all the vocabulary of a language. To solve this problem, we calculate probabilities based on a history of reduced size model called the n-gram. An n-gram is a subsequence of n items constructed from a given sequence. From a given sequence of words it is possible to obtain the likelihood of the occurrence of the next word function. From a training corpus, it is easy to construct a probability distribution for the next word with a history of size n [28]. This modeling is actually an order Markov model n where only the last n observations are used to predict the next word. And a bi-gram is an order Markov model 2 [29].

Language modeling means finding the most likely word knowing those preceding it. This task is performed during the training phase of the corpus of the target language. There are two tool boxes commonly used for building language models, the SRILM tool [30] and the tool CMU-Cambridge [31]. These two jewels have open access; and similar features.

3.3 Methods of Error Detection

A natural method to simplify the design of a dictionary is to consider a corpus of documents, assuming that all documents of this corpus are spelled correctly, simply collect all the words used to automatically create the dictionary and carry out a cleaning step. We exploited and downloaded various freely available sources of information (texts and Arabic dictionaries). Then, we cleaned the text and added to the dictionary: it is a file that will contain the maximum of words in Arabic [5].

In this section we describe the principle of detecting misspelled words, spell checking, and sorting of possible corrections. We used a simple approach to detect misspelled words based on the use of a dictionary: a word of text is considered misspelled when it does not appear in the dictionary. If the data volume is large enough, we can hope to cover enough cases to get a useful system. Each of the words of the text is compared to words in the lexicon.¹ Any word not found in the lexicon is considered wrong. To detect misspelled words, just browse the dictionary. We transliterated into Latin dictionary as a transliteration representing the Arabic characters as Latin characters because the handling of Arabic characters is difficult.

To browse the dictionary and detect errors we used the binary search because its use has been extremely efficient compared to conventional sequential search. Then we organized our dictionary as follows to make the quick search: we currently cut The dictionary into dictionaries following the length of each word ie words that have the same length are grouped in a dictionary with a noun that carries the length of the words that compose it.

3.4 Spelling Correction: Use of Edit Distance to Order Suggestions

For a misspelled word we can add spaces in the middle of it to break it down into words. Our first prototype will analyze the words obtained and it will search in our data dictionary. If the words exist in the latter, they see them as a correction proposal.

The measurement of the best known distance², the Levenshtein distance is a simple metric between two channels, or each operation at a cost of 1 [7]. Levenshtein distance can be weighted by the length of the compared strings: the score is then divided by the sum of the lengths of the two compared strings. This is a distance in the mathematical sense, so in particular it is a positive number or zero, and two strings are identical if and only if their distance is zero. Calculate the minimum number of operations needed to transform one string characters in to another, where an operation is defined as the insertion, deletion, or substitution to move from one channel to another. One application of this distance is spellchecking: when a person types in a word, compared to a dictionary. If the word is present, nothing is done, otherwise, there are attempts from dictionary words [5], those whose Levenshtein distance to the typed word is less than a given limit. The nearest words are suggested as replacement first. The measurement of the Levenshtein distance between two strings (String1 and String2) consists of implementing the following algorithm:

¹ A glossary of animal words constituted the canonical forms (lemmas or bases), proclitics, prefixes, suffixes and enclitic, and a morphological analysis algorithm.

² http://en.wikipedia.org/wiki/Levenshtein_distance

Table 1. Transliteration of Arabic characters as Latin characters

Arabic characters	Characters: pronunciation	Transliteration into Latin letters	Arabic characters	Characters: pronunciation	Transliteration into Latin letters
ء	hamzā	ʾ	ف	Fā'	f
ا	ālif	ā	ق	qāf	q
ب	Bā'	b	ك	kāf	k
ت	Tā'	t	ل	Lām	l
ث	Tā'	ṭ	م	mīm	m
ج	Jīm	ǧ	ن	nūn	n
ح	ḥā'	ḥ	ه	Hā'	h
خ	Xā'	ḫ	و	wāw	w / ū
د	dāl	d	ي	Yā'	y / ī
ذ	ḏāl	ḏ	ا	fathā	a
ر	Rā'	r	ا	ḏammā	u
ز	zāy	z	ا	kasrā	i
س	Sīn	s	ا	tanwīn	ā / an
ش	Sīn	š	ا	tanwīn	ū / un
ص	ṣād	ṣ	ا	tanwīn	ī / in
ض	ḏād	ḏ	ة	tā' marbūṭā	ā (at en annexion)
ط	tā'	t	ى	Alif maqṣūrā	ā
ظ	ẓā'	ẓ	آ	Alif mamdūdā	ā
ع	'ayn	ʿ	أ	hamzā	á
غ	ǧayn	ǧ	إ	hamzā	ì
			إ	hamzā	í
			ؤ	hamzā	w
			ئ	hamzā	y
			ـ	sukun	
			ـ	šaddā	lettre redoublée

Function 1.

Int LevenshteinDistance(char str1[1..lenStr1], char str2[1..lenStr2])

```
declare int d[0..lenStr1, 0..lenStr2]
declare int i, j, cost
for i from 0 to lenStr1
    d[i, 0] := i
for j from 0 to lenStr2
    d[0, j] := j
for i from 1 to lenStr1
    for j from 1 to lenStr2
        if str1[i] = str2[j] then cost := 0
        else cost := 1
        d[i, j] := minimum (
            d[i-1, j ] + 1,      // deletion
            d[i , j-1] + 1,      // insertion
            d[i-1, j-1] + cost   // substitution
        )
    if (i > 1 and j > 1 and str1[i] = str2[j-1] and str1[i-1] = str2[j]) then
        d[i, j] := minimum (
            d[i, j],
            d[i-2, j-2] + 1      // transposition
        )
return d[lenStr1, lenStr2]
```

Search word candidate for the correction is made with the edit distance reversed as follows: First, all words with an edit distance equal to 1 with the wrong password is generated by applying the editing operations that is the insertion, deletion, substitution and transposition. Then, each word is searched previously generated in e sorts or hash. If there is, then it is retained as a possible correction of the erroneous word.

Sorting candidates corrections takes into account the extent used in the selection step, as well as statistical measures. The choice of the most probable correction is done by giving each candidate a score Correction. The lower the score, the more likely it is that the candidate correction is the correct spelling of the word to correct.

Several ways to define this score are possible. The chosen solution is to set the score as the number of occurrences of the candidate correction in the text corpus more frequently correcting a candidate appears, it is more likely. This applies to sorting out words assigns a distance of levenshtein low enough to misspelled words.

4 Evaluation of the Prototype of Orthographic Correction

The corpus studied consists of a set of journalistic articles published by the newspaper "Le Monde Diplomatique" in its Arabic version. This source has the advantage of providing large quantities of good quality text The topics addressed are fairly general and treat various themes of the policy world news, economic, cultural,

sports, etc. This corpus contains 1009 items, accumulating a total of 4,126,631 graphic words grouped under 322,156 different forms. Using a wide variety of themes and addressed areas aims to have a broad coverage of the words of the language. We used a corpus containing 164 Arabic texts collected in UTF-8 format.

The evaluation system is a crucial step. It helps to highlight its strengths and limitations, and to find leads for possible improvement. To get an accurate comparative assessment of our system, we chose different sizes of texts, each containing a varied set of misspelled words by comparing with other spell checkers nowadays, virtually present in all computer applications where the text is called to be entered by the user. This is usually notified of an incorrect entry with a red underline the wrong word. Examples of such applications are: the word processing software, email clients, source code editors and programming environments, internet search engines. The causes of the error are manifold and we find more than one way to classify [28].

Evaluation of any information retrieval system based on the calculation of a set of metrics. These calculations used to assess the proportion of errors displayed by the system from the ideal result. The metrics typically used are:

- **Number of words:** the number of words of each text.
- **Number of real errors:** the number of erroneous words of each text.
- **Number of errors detected:** the number of errors detected by the system.
- **Precision (P)** is an assessment of system noise. It measures the proportion of relevant system responses among all answers he provided
- **The recall (R)** is an assessment of the coverage of the system. It measures the amount of a relevant system compared to the number of responses ideal answers.
- **The F-measure (F)** is a metric that combines in a single value precision measurements and return to penalize excessive inequalities between the two measures.

However, given the boundary problems, we had to redefine the evaluation parameters to account for partially correct answers [32]. These valuation parameters become:

$$\text{Precision: } P = \frac{\text{number of errors correctly (partially correct + incorrect) detected}}{\text{number of identified errors}}$$

$$\text{Recall: } R = \frac{\text{number of errors correctly (partially correct + incorrect) detected}}{\text{actual number of errors}}$$

$$\text{F-measure: } F = \frac{2 \cdot P \cdot R}{P + R}$$

Table 2. Experience with our system.

Text	Text 1	Text 2	Text 3	Text 4	Text 5	Text 6
Number of words	100	80	60	70	50	40
In real errors Nb	33	27	23	17	13	11
Nb errors	23	20	19	12	11	9
Precision	69,69 %	74,07 %	82,60 %	70,58 %	84,61 %	81,81 %
Recall	71,23 %	83,37 %	81,49 %	78,14 %	85,75 %	80,41 %
F-measure	70,45 %	78,44%	82,04 %	74,16 %	85,17 %	81,10 %

Table 3. Experience with Word 2007

Text	Text 1	Text 2	Text 3	Text 4	Text 5	Text 6
Number of words	100	80	60	70	50	40
In real errors Nb	33	27	23	17	13	11
Nb errors	22	19	18	12	10	8
Precision	66,66 %	70,37 %	78,26 %	70,58 %	76,92 %	72,72 %
Recall	71,04 %	81,17 %	79,42 %	77,96 %	84,47 %	79,63 %
F-measure	68,78 %	75,38%	78,83 %	74,08 %	80,51 %	76,01 %

The three measures commonly used to assess a spelling correction system are the recall rate R, precision P and that of F. Measures to do this, and after the assumption of texts were aligned, the system begins its analysis sentence by sentence. The tests that are applied are encouraging. The summary evaluation we conducted shows that Arabic spellchecker gives in most proper words. The results of the evaluation show a gain of our method compared to the spellchecker "word 2007". After viewing these experiences, we see that these results are closer and are a good starting point for further research to improve our first spellchecking prototype.

5. Conclusion and Outlook

In this work we presented in detail the architecture of a first prototype for spelling Arabic. First, we describe the resources to spell checking and the implementation of this module. Then we evaluated our first prototype. A preliminary performance evaluation was conducted, which helped highlight the difficulty of the task and identify some of the current system limits. We can not certainly say that our work is complete when several improvements can be made. But we hope at least that we managed to achieve a simple spell check system.

As perspective of this work improvements can be made on the resources made increasing the dictionary size to cover a maximum of the Arabic language. Other techniques may be explored as ngram techniques and probabilistic. Semantic type errors can be a future project, too.

Referenes

1. Debili, F., Achour H., Souici E.: La langue arabe et l'ordinateur: de l'étiquetage grammatical à la voyellation automatique, Correspondances de IRMC, N°71, pp. 10-28, (2002).
2. Peterson, L.: Computer Programs for Detecting and Correcting Spelling Errors. Comm. ACM, 23 (1980).
3. Mitton, R.: Ordering the suggestions of a spellchecker without using context, Natural Language Engineering, (2009)
4. Kukich, K.: Techniques for Automatically Correcting Words in Text. R. Mitton, Ordering the suggestions of a spellchecker without using context, (2009).

5. Bacha, K., Zrigui, M.: Design of a Synthesizer and a Semantic Analyzer's Multi Arabic, for use in Computer Assisted Teaching, *International Journal of Information Sciences and Application* , IJISA , pp. 11-33, (2012).
6. Enguehard, C., Naroua, H.: Evaluation of Virtual Keyboards for West -African Languages. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)* , Marrakech, Morocco, 28-30 (2008).
7. Levenshtein.V.: Binary codes capable of correcting deletions, insertions and reversals, (1966).
8. Kukich, K.: Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24 (1992).
9. Peterson, J.: Computer Programs for Detecting and Correcting Spelling Errors. *Comm. ACM*, 23 (1980).
10. Pollock, K., Zamora, A.: Automatic Spelling Correction in Scientific and Scholarly Text, (1984).
11. Oflazer, K.: Error-tolerant Finite-state Recognition with Applications to Morphological Analysis and Spelling Correction, (1996).
12. Ben Othmane, Z., Zribi, A.: Algorithmes pour la correction orthographique en arabe, (1999).
13. Shaalan, K., Allam, A., Gomah, A.: Towards automatic spell checking for arabic. In *LanguageEngineering*, (2009).
14. Haddad, B., Yaseen, M.: Detection and correction of non-Words in Arabic: A hybrid approach. *International Journal of Computer Processing of Oriental Languages*, (2007).
15. Zerrouki, T., Balla, A.: Implementation of indexes and circumfixes in the spellcheckers. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, (2009).
16. Muaidi, H.: Extraction Of Arabic Word Roots: An Approach Based on Computational Model and Multi-Back propagation Neural Networks. PhDthesis, DeMontfort University-UK", (2012).
17. Gueddah.H., Yousfi, A., Belkasmi, M.: Introduction of the weight edition errors in the Levenshtein distance, *International Journal of Advanced Research in Artificial Intelligence*, Vol 1 Issue 5, pp 30-32, Aout (2012).
18. Ndiaye, M., Vandeventer, F.: A Correcteur orthographique apprentissage du français. *BULAG*, 29:117-134 . (2004).
19. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals, (1966).
20. Yannakoudakis, E., Fawthrop, D.: The rules of spelling errors. *Information Processing and Management*, 19, 87-99 (1983).
21. Emirkanian, L., Louisette, N., Lorne, H.: La correction des erreurs d'orthographe d'usage dans un analyseur morpho-syntaxique du français. *Langue française*, 106-122 (1989).
22. Heer, T. The application of the concept of homosemy to natural language information retrieval. *Information Processing and Management*, 18(5), 229-236 (1982).
23. Angell, A., Richard C., George E.: Peter Automatic spelling correction using a trigram similarity measure. *Information Processing and Management*, 19(4), 255-261 (1983).

24. Lefevre P. Caillaud N., "Logiciel d'accès par voisinage à un dictionnaire automatique du français courant", (1992).
25. Sinah , R.: On partitioning a dictionary for visual text recognition, (1990).
26. Mazal, L., Vidal, E.: Computation of normalized edit distance and applications, (1993).
27. Stolcke, A.: SRILM An Extensible Language Modeling Toolkit. Proc. ntl. Conf. on Spoken Language Processing , (2002).
28. Merhbene, L., Zouaghi, A., Zrigui, M.: Arabic Word Sense Disambiguation.", ICAART (1) , 652-655 (2010)
29. Bacha, k., Zrigui, M.: Designing a Model of Arabic Derivation, for Use in Computer Assisted Teaching. KEOD (2012)
30. Bacha, K., Zrigui, M.: Morphological Analysis in the Environment "TELA". SCSE , 521-528 (2015).
31. Clarkson, M.: Statistical Language Modeling Using the CMU-Cambridge Toolkit. In Proceedings of EuroSpeech (1997).
32. Suzan, V.: Context-sensitive spell checking based on word trigram probabilities. Master thesis (2002).
33. Cunningham, H., Bontcheva, K.: Named Entity Recognition, Actes de la conférence internationale RANLP 2003, Borovets, Bulgaria, (2003).

Emerging Technology Enabling Dyslexia Users to Read and Perceive Written Text Correctly

Tereza Pařilová, Filip Mrváň, Bruno Miřík and Eva Hladká

Masaryk University, Faculty of Informatics, Department of Computer Systems and Technologies, Brno, Czech Republic
{parilova,409943,410181,eva}@mail.muni.cz

Abstract. Dyslexia is treated by many specialists as a cognitive impairment involving visual attention deficit. It may cause an image of letter rotation or overlapping. Both children and adults suffer from this disease, differing in individual needs and seriousness. With a growing amount of information being distributed digitally, there is a need to accommodate online text to dyslexic users. However, with different types of operating systems, web browsers and substantial dyslexia individualism, it is not easy to fully automatize such needs. We are developing an extension for Chrome browser which is based on our previous cognitive research and empirical data. Such an extension will allow users with dyslexia to accommodate web content with special fragmentation sign which demonstrably suppresses reading problems caused (not only) by dyslexia.

1 Introduction

Depending on the language examined and the statistical methods used, it is stated that 5 % to 20 % of people worldwide suffer from dyslexia and other reading disorders, although only some of them are diagnosed [1]. Dyslexia is categorized in the International Classification of Diseases (ICD) as a symbolic dysfunction and reading disorder [2]. Categorized as mentioned, dyslexia is the primary neurological cognitive disorder affecting brain parts and visual pathways [3]. The requirements of textual accessibility for people with dyslexia of various languages has not yet been properly established. The level of dyslexia problems is very much dependent on the profile of the language. The main problems that users with dyslexia deal with are letter reversal (especially mirror reverse) and lateral masking (overlapping and losing adjoining characters).

Accessibility to any kind of information available digitally nowadays is a key factor to the equality of rights, breaking down social barriers. Considering that study materials and daily news are very much accessible through the Internet, accommodating digital content to diversely disabled users must be seen as a natural thing.

Dyslexia is typical for its problem variability, or rather, individuality. Therefore, it is not easy to find a general solution for every user, it is quite a tangible and unbounded task for satisfying technical solutions. Also, people are dealing with reading problems for instance after brain surgery, brain tumors, epilepsy and so on. The need to accommodate text for different groups of people with reading difficulties is therefore crucial. To make graphemes easier to read and distinguish between the original and the reversed

grapheme, we are studying text processing ways and come up with new solutions. This paper describes an autonomous unit of such a solution – an extension designed for Chrome web browser, based on our previous empirical studies.

1.1 Phonetic and Anti-Phonetic Languages Restrictions

Phonetic languages have simple rules of pronunciation as opposed to non-phonetic languages. Languages that can be marked as phonetic are Czech, Slovakian, Finnish, less phonetic ones are Swedish, Norwegian, Turkish or Albanian. Some of the non-phonetic languages are for instance English or French [4]. To distinguish between more or less phonetic languages, e. g. language recognition system can be used together with phoneme orthography evaluation data (Fig. 1) [5].

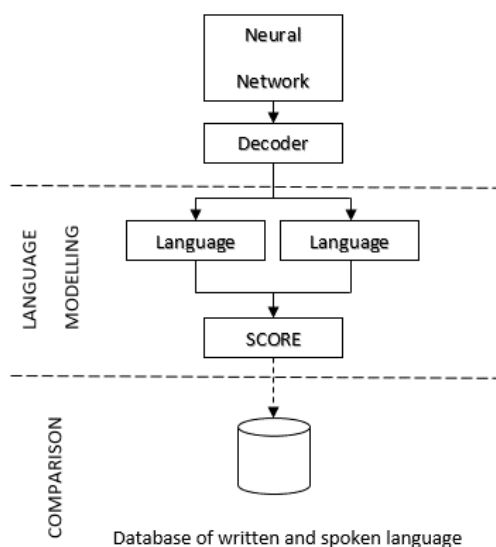


Fig. 1. Language model of phoneme recognition [5] adapted for phoneme-orthographic evaluation of languages and for further testing of usability of text modification hypothesis.

This is only one of the many reasons why different languages need different methods and approaches to the dyslexia solution and cannot apply general rules to all languages.

We are able to test our solution for dyslexic users also on the mentioned languages, as they contain problematic letters, which are pronounced phonetically (Table 1).

2 Related Work

To the best of our knowledge, there is no tool for accommodating web text as sophisticated as our proposed tool. Solutions used by dyslexic users are usually simple but not as interactive as necessary, not solving the real problems of dyslexia.

Table 1. The summary of languages and words with problematic (doubled) graphemes and its phonetic pronunciation – problematic letters do not change the pronunciation (ss→s etc.).

Language	Word	Pronunciation	English
Turkish	passerby	[pasrbai]	passerby
Albanian	hallkë	[halke]	ring
Dutch	sprookje	[sprokie]	fairy tale
Swedish	berättelse	[berætelse]	tale
Norwegian	utdannelse	[u:tdanelse]	education
Finnish	tarkoittaa	[tarkoita]	mean

2.1 Plain text

One of the possible solutions tested within a group of people with dyslexia in the English language is the use of plain text with no graphical elements that may distract the user. The plain text also deals with font size and type (serif vs. sans serif) [6]. It is easy to change the font in some types of documents but as for technical aspects, still not every dyslexic user is able to change the font or disturbing visualization of a PDF or a web page.

2.2 Speech synthesis

Speech synthesis is often used to convert text to audio format [7]. There are studies which address this problem and propose solutions mainly for English [8, 9] and Spanish [10]. However, none of these solutions accommodate all aspects of dyslexia, moreover, due to grammar and word complexity of some languages, these applications are not helpful in specific societies. These systems are very much used in technologies helping blind or visually impaired people. A system that detects “difficult words” from a database and offers their synonyms was introduced in Spain and is being used in different platforms (for laptops, mobile phones, tablets etc.) [11].

2.3 Text4All

Text4All¹, designed by Topac [12] is a web page where a user types the url which he/she wants to visit and it restricts most of the graphics and structures the text in a very simple way. It may in some cases help a user to remove elements that are too disturbing (e. g. too many graphics) and make the text heavier (bigger letters, blanks etc.) but it may also eliminate important parts of such web pages (some graphic elements are crucial in study materials, instructions and even in daily news, Fig. 2). Because of changing the font, the web page sometimes causes letter overlapping which is unfortunately the problem that dyslexia causes (Fig. 3).

¹ <http://www.text4all.net/dyswebxia.html>

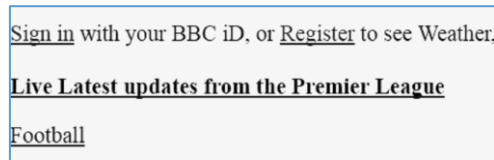


Fig. 2. Web page of BBC accommodated by Text4All – no pictures with articles.²

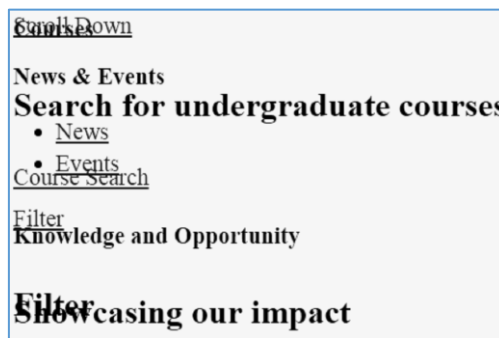


Fig. 3. Web page of University of Leeds accommodated by Text4All – overlapping text.³

2.4 Fox Replace

The extension for Mozilla Firefox permits changes in web page text by a manual insertion of any machine readable symbol - such an extra symbol is then always inserted into a specified place anywhere in a text (e. g. *i* between *m* and *n*). The problem with such an extension is that the rule that users manually set is applied to all content, including web page links. The result is that the link reports an error (Fig. 4).

2.5 Open Dyslexic

This is well-known extension at first built for Mozilla Firefox. Now, it is available for Chrome and Android as well. The idea is based on thickening the grapheme's curve to make confusing letters more distinguishable.

As stated before, the main problem of dyslexia dwells in visual attention, making the similar letters overlap, rotate etc. Thus, thickness of letters does not solve the problem. Moreover, the same parts of very similarly looking letters are thicker or thinner. Therefore, such text can still be hardly readable. Also, users that tried Open Dyslexic font have quite divergent opinions arising from their individual needs. "*The issue I have*

² http://www.text4all.net/do?url=www.bbc.co.uk&styleTextSize=22&op=dyswebxia2&styleTextColor=000000&styleBackground=FAFAC8&colors=000000_FAFAC8&styleTextFont=original&layout=original&lang=EN&langOp1=DifficultWords&langOp2=replaceNumbers

³ http://www.text4all.net/do?url=www.leeds.ac.uk&styleTextSize=22&op=dyswebxia2&styleTextColor=000000&styleBackground=FAFAC8&colors=000000_FAFAC8&styleTextFont=original&layout=original&lang=EN&langOp1=DifficultWords&langOp2=replaceNumbers

with OpenDyslexic is that it has the bottom heavy theory which I feel does not do anything. You'll still flip the P and Q because they look so similar.”, “. . . it was a relief to get back to a standard font after trying it out.”⁴



Fig. 4. Extra symbol is being automatically inserted into web address which results in Error⁵.

2.6 A Context-Aware Synonym Simplification

This extension for Chrome was meant to be a method that generates more simple synonyms of problematic words based on the context of the sentence. The database of synonyms was taken from Google Books N-gram Corpus and Open Thesaurus and presented in [13]. Authors argue that the extension, due to language mutations of the Corpus and Thesaurus, is easily transferable to other languages. Based on the published example (Fig. 5), some simplification might be misleading for the reader to understand the context properly. This can cause problems especially to students when reading study materials or important information of specific disciplines. Since the authors of the paper presenting CASSA tested this approach with only three volunteering subjects and do not present an argument of the functionality in other languages, it is questionable how practical this extension is in real use.

3 Reading Simplification Based on Grapheme Fragmentation

Our research focuses on building a huge fully-automatized and highly personalized system that would make digital text more accessible to people with dyslexia and other reading disorders. In this paper we present one part of such a system that is responsible

⁴ <https://nycdyslexiaresearch.wordpress.com/2012/10/12/is-open-dyslexic-font-actually-easier-to-read/>

⁵ www.bbc.co.uk/acc-es-sibi-l-ity, originally www.bbc.co.uk/accessibility

for the modification of web text. It is a Chrome browser extension and serves as an external part of the whole system, cooperating with its back-end (Fig. 6).

Original	El contemplaba en silencio aquella cruz. <i>He was contemplating in silence that cross.</i>
Baseline	Él veía en silencio aquella cruz. <i>He was seeing in silence that cross.</i>
LexSis	Él consideraba en silencio aquella cruz. <i>He was considering in silence that cross.</i>
CASSA	Él miraba en silencio aquella cruz. <i>He was looking in silence that cross.</i>

Fig. 5. Chosen words simplified by three different algorithms [13].

The main idea involves changing the visual perception of web pages on the user side. A hypothesis that inserting a fragmentation sign may help dyslexic users (both with developmental and secondary dyslexia) was preliminarily studied in Pařilová [14, 15] and Pařilová, Hladká and Bayer [16]. On ten volunteering subjects it shows that fragmentation has a potential to improve speed of reading while understanding of context is not disturbed (Tab. 2). There are no other works studying text fragmentation for dyslexic users but for instance larger spacing theory shows some improvement in reading, except in extensive texts it lacks of visual simplicity and intelligibility. Thus, the motivation for our work is to draw from empirical theories and establish innovative and significantly individual way of text accommodation.

User	Original/seconds	Modification/seconds
1	17,73	17,06
2	17,98	17,16
3	18,31	18,06
4	20,07	19,34
5	19,89	19,11
6	17,63	16,98
7	17,92	17,36
8	18,47	18,01
9	20,08	19,66
10	19,75	19,03

Table 2. Results of speed reading (and comprehension) testing [14].

The programming part of our extension is built on JavaScript. The extension works as follows.

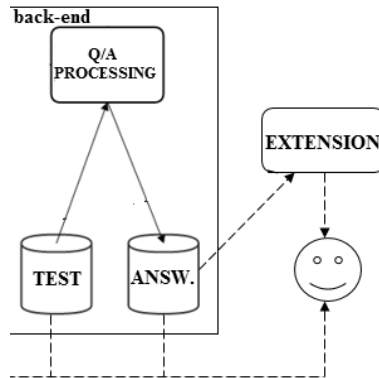


Fig. 6. Cooperation architecture between emerging system, the extension and a user.

3.2 Modification Process

To find the dyslexic user’s needs to accommodate text, we use sets of tests available from the system (containing a database of similar tests). When a particular ID is sent to the server with a completed test, the server responds and the result is a matrix of letters that need to be divided. The “need” is set to a portion of errors made in test (e. g. *do divide top 30 % of errors made in the test*, Tab. 3).

The tests look like the following examples (Fig. 7, 8 and Tab. 4).

Table 3. Percentage of user’s error possibly made in tests. Based on above results and set percentage the extension will modify patterns, e. g. *do* and *mn*.

Errors	%
mn	73
do	51
ce	24
tl	1



Fig. 7. Example of the test: Please mark the letter that is reversed.

H H
p p
m n
n u
r u
H N
y y

Fig. 8. Example of the test: please mark a pair of letters that seem to you not identical in the pair.

Table 4. Example of the test: Please find a word “mnul” (mark all 4 letters).

m	n	u	i	h
n	m	h	n	l
u	i	m	l	h
l	m	l	k	n
k	h	u	m	i

Jquery.min.js contains a JavaScript storage jQuery which is used for letter pattern searching and is implemented in background.js. The extension works on both http and https.

There are several ways to search for a grapheme pattern in a text, like SHIFT-OR algorithm, Aho-Corasick algorithm or Levenstein distance. However, we simply use regular expression advantages – fast processing, simplicity of use, possibility of case sensitivity adjustment, not too big dataset etc.

Regular expressions search for a defined letter pattern and divide it by the mentioned fragmenting sign – a dash. The searching runs only inside tags, tags themselves are not searched so that the visual shape of a piece of text changes but the content does not (Fig. 9, Fig. 10). Therefore, any text, including hypertext, can be more readable for the

user but the content of the page stays unchanged and after browsing any link and sub link is working properly (in contrast with Fox Replace).

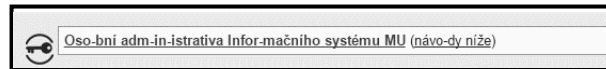


Fig. 9. Web page link after extension modification of Czech text.

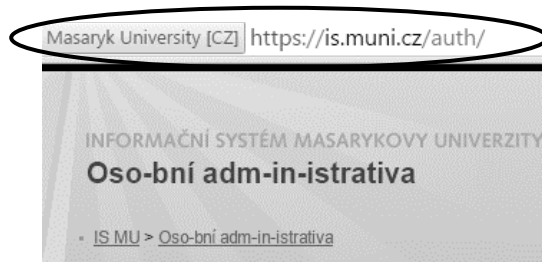


Fig. 10. Correctly functional web page with no modification of the address.

Text in Finnish after the fragmentation would visually look like on Fig. 11, in Turkish on Fig. 12, in Bulgarian on Fig. 13, in Norwegian on Fig. 14 and in Swedish on Fig. 15.

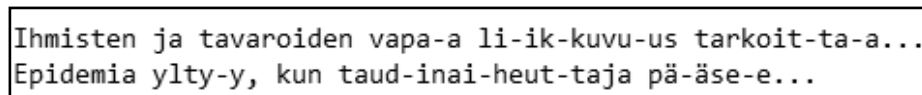


Fig. 11. Part of Finnish text from university website.⁶

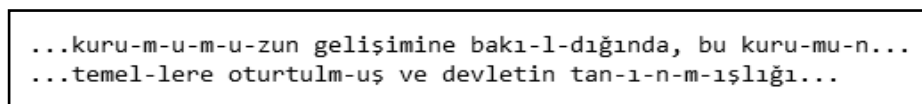


Fig. 12. Part of Turkish text from Ministry of foreign affairs website.⁷

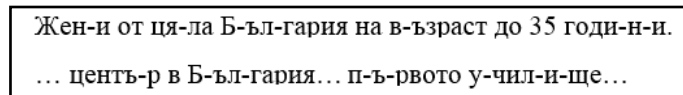
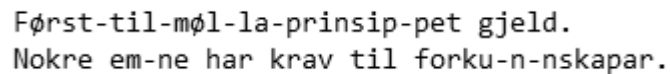


Fig. 13. Part of Bulgarian text from university web page.⁸

⁶ <https://www.helsinki.fi/fi/uutiset/epidemiati-tulevat-puskista>

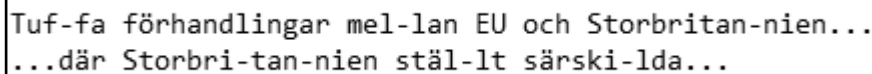
⁷ <http://diab.mfa.gov.tr/tr/hakkimizda/bakan/>

⁸ https://www.uni-sofia.bg/index.php/bul/universitet_t



Først-til-møl-la-prinsip-pet gjeld.
Nokre em-ne har krav til forku-n-nskapar.

Fig. 14. Part of Norwegian text from university website.⁹



Tuf-fa förhandlingar mel-lan EU och Storbritan-nien...
...där Storbri-tan-nien stäl-lt särski-lda...

Fig. 15. Part of Swedish text from Ministry of affairs web pages.¹⁰

Conclusion

Dyslexia is a part of the cognitive disease group that affects both children and adults of any age. It can manifest after a surgery, tumors or brain injuries and can be temporary. However, most people with dyslexia appear to have lifelong lasting problems with decoding written text. For children it is necessary to train reading, because their brains are more plastic and accommodate much faster than adults' brains and their ability to read better and faster is expectable. But the same advice is true for adults – training is the only option to help improve reading. There were several attempts to use technologies for other impairments, for instance text to speech systems and screen readers [13], and also attempts to develop technology aimed at dyslexia. For web page reading, it is especially Fox Replace, the extension for Mozilla browser, Text4All web service and CASSA, the algorithm for simplifying sentences by using synonyms. However, none of the mentioned solutions is very comfortable or easily editable over time. The more information we find on the Internet (daily news, study materials, travel information, e-mail online box etc.), the more we need such a solution.

Our Chrome browser extension, autonomous part of an emerging service, focuses on individual needs and personalization of web pages. It is able to deal with transformation of one's need over time. It allows the user to use the extension on any computer as the extension is easily switchable and needs only one download. There is no need to sign up for the service which can be very much appreciated by its users (due to security of their personal data). This extension can be transferred to different (at least a little) phonetic languages. The extension is a challenge for further design and development of dyslexia assistive technology and offers the most up to date help that users with dyslexia may need while being online.

⁹ <http://www.uio.no/studier/admin/eksamen/tilrettelegging/>

¹⁰ <http://www.regeringen.se/>

Acknowledgements

The research and project work was done thank to “Representation of Czech Republic in European Research Consortium for Informatics and Mathematics” project ERCIM-CZ sponsored by the Ministry of Education, Youth and Sports, and with the support of the University Development fund FRVS/0430/2001 of the Masaryk University.

References

1. de Santana, V. F., de Oliveira, R., Dell Anhol Almeida, L., Calani Baranauskas, M. C.: Web accessibility and people with dyslexia: a survey on techniques and guidelines. Proceedings of the International Cross-Disciplinary Conference on Web Accessibility. Article No. 35. ACM (2012)
2. International Classification of Diseases. World Health Organisation (2015)
3. Danelli, L., Berlingeri, M., Bottini, G., Ferri, F., Cacchi, L., Sberna, M., Paulescu, E.: Neural Intersections of the Phonological, Visual Magnocellular and Motor/Cerebellar Systems in Normal Readers: Implications for Imaging Studies on Dyslexia. *Human Brain Mapping*, 34(10), pp. 2669-2687 Wiley (2012)
4. Bohn, O. S., Munro, M. J.: *Language Experience in Second Language Speech Learning: In Honor of James Emil Flege*. John Benjamins Publishing (2007)
5. Matějka, P., Schwarz, P., Burget, L., Černocký, J.: Use of Anti-Models to Further Improve State-of-the-Art PRLM Language Recognition System. Proceedings of ICASSP '06. pp. 197-200 IEEE (2006)
6. Rello, L., Baeza-Yates, R.: Good Fonts for Dyslexia. Proc. of ASSETS '13. Article No. 14 ACM (2013)
7. Henton, C.: *Text-to-Speech Synthesis Development*. The Encyclopedia of Applied Linguistics. Blackwell Publishing Ltd. (2012)
8. Kanvind, G., Rello, L., Baeza-Yatez, R.: IDEAL: a dyslexic-friendly eBook Reader. Proceedings of ASSETS '12. pp. 205-206 ACM (2012)
9. Georgopoulos, V. C., Malandraki, G. A., Stylios, Ch. D.: A fuzzy cognitive map approach to differential diagnosis of specific language impairment. *Artificial intelligence in Medicine*. pp. 261-278 Elsevier Science Publishers Ltd. (2003)
10. Viot, G.: Fuzzy logic in C, Dr. Dobb's Journal. 18(2), pp. 40-49 (2003)
11. Bott, S., Rello, L., Drndarevic, B., Saggion, H.: Can Spanish Be Simpler? LexSiS: Lexical Simplification for Spanish. *Coling '12*. pp. 357-374 ACL (2012)
12. Topac, V.: *Text4All Web Service*. Polytechnic University of Timisoara, Romania (2013)
13. Baeza-Yates, R., Rello, L., Dembowski, J.: CASSA: A Context-Aware Synonym Simplification Algorithm. *NAACL-HLT '15*. pp. 1380—1385, ACL (2015)
14. Pařilová, T.: Graphical Modification of Text. An Approach To Dyslexic Users. *SLPAT '14*. pp. 9-12, ACL (2014)
15. Pařilová, T. : Making Text Content Accessible for Users with Dyslexia in Czech. *ECAET '14*. European network for the integration of Web 2.0 (2014)
16. Pařilová, T. Hladká, E. And Bayer, J. : Customizing Individual Needs of Accessing Text Information for Users with Dyslexia. *European Brain and Behavior Society Join Meeting* (2015)

Reviewing Committee of the Volume

Ajith Abraham	Sanda Harabagiu	Maria Teresa Paziienza
Bayan Abushawar	Yasunari Harada	Ted Pedersen
Hanady Ahmed	Ales Horak	Viktor Pekar
Yeşim Aksan	Nancy Ide	Anselmo Peñas
Hend Alkhalifa	Diana Inkpen	Soujanya Poria
Adil Alpkocak	Aminul Islam	Marta R. Costa-Jussà
Rania Al-Sabbagh	Guillaume Jacquet	Ahmed Rafea
Galia Angelova	Miloš Jakubiček	Allan Ramsay
Marianna Apidianaki	Doug Jones	Fuji Ren
Aladdin Ayesb	Sylvain Kahane	German Rigau
Alexandra Balahur	Bahar Karaoglan	Fabio Rinaldi
Sivaji Bandyopadhyay	Dimitar Kazakov	Horacio Rodriguez
Leslie Barrett	Imtiaz Khan	Paolo Rosso
Roberto Basili	Alma Kharrat	Vasile Rus
Anja Belz	Tarik Kisla	Franco Salvetti
Christian Boitet	Philipp Koehn	Rajeev Sangal
Igor Bolshakov	Leila Kosseim	Kepa Sarasola
Karim Bouzoubaa	Senem Kumova Metin	Fabrizio Sebastiani
Cem Bozsahin	Mathieu Lafourcade	Nasredine Semmar
Nicoletta Calzolari	Bing Liu	Khaled Shaalan
Nick Campbell	Cerstin Mahlow	Altynbek Sharipbayev
Fazli Can	Suresh Manandhar	Serge Sharoff
Michael Carl	Diana Mccarthy	Bernadette Sharp
Violetta Cavalli-Sforza	Alexander Mehler	Grigori Sidorov
Niladri Chatterjee	Farid Meziane	Kiril Simov
Ilyas Cicekli	Rada Mihalcea	John Sowa
Dan Cristea	Evangelos Milios	Efstathios Stamatatos
Samhaa El-Beltagy	Ruslan Mitkov	Dzhavdet Suleymanov
Michael Elhadad	Dunja Mladenic	Maosong Sun
Gülşen Eryiğit	Hermann Moisl	Jun Suzuki
Aly Fahmy	Masaki Murata	Stan Szpakowicz
Anna Feldman	Preslav Nakov	Hristo Tanev
Alexander Gelbukh	Costanza Navarretta	William Teahan
Dafydd Gibbon	Nicolas Nicolov	J.-M. Torres-Moreno
Roxana Girju	Joakim Nivre	George Tsatsaronis
Gregory Grefenstette	Kjetil Nørvåg	Olga Uryupina
Ahmed Guessoum	Attila Novák	Manuel Vilares Ferro
Tunga Gungor	Nir Ofek	Aline Villavicencio
Hatem Haddad	Farhad Oroumchian	Piotr W. Fuglewicz
Kais Haddar	Partha Pakray	Marilyn Walker
Lamia Hadrich Belguith	Ivandre Paraboni	Andy Way
Eva Hajicova	Saint-Dizier Patrick	Bonnie Webber

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
en septiembre de 2016
Printing 500 / Edición 500 ejemplares