Advances in Opinion Mining, Social Network Analysis, and Authorship Attribution

Research in Computing Science

Series Editorial Board

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 110,** abril 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 110**, April 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 110

Advances in Opinion Mining, Social Network Analysis, and Authorship Attribution

Alexander Gelbukh (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

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

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

This special issue of the journal *Research in Computing Science* is devoted to closely interrelated topics of opinion mining, social network analysis, and authorship attribution.

Opinion mining is an emergent and very active area at crossroads of natural language processing, artificial intelligence, computer science, and big data analysis. The main goal of an opinion mining task is to collect statistics of crowdsourced opinions on a given topic, such as a product, a service, an event, a political entity, etc., addressing questions such as "do users like iPhone 7?", "will Trump win the 2016 US presidential elections?", or "what is the best hotel in Singapore?", basing on the opinions of thousands or millions of people, typically available in Internet via social networks.

Collecting such opinions is very useful for many purposes. Opinion mining helps businesses to increase their income by better addressing the needs and preferences of their customers. It helps governments and political parties to better understand popular opinion and adjust their actions accordingly, winning more votes. Most importantly, it improves the life quality of ordinary people by providing recommendations based on experience of other consumers in order for us to make more informed buying decisions. It also enables real-time democracy: the ability of citizens to affect the actions of the government right now, without waiting for the next elections.

Technically, opinion mining systems rely on analysis of sentiments and emotions expressed in user-generated texts or video clips. Some authors, notably Bing Liu, consider the term "opinion mining" to be synonymous with "sentiment analysis." However, I personally consider that these are two different processes: sentiment analysis is the process of analyzing a given single text with the purpose of determining the sentiment polarity, emotion, or opinion expressed in this text, while opinion mining should refer to the process of aggregating the results of analyzing a large number of individual documents into a statistical report such as majority opinion, breakup by social groups, etc. Thus, a number of papers included in this volume correctly refers to the task they address as sentiment analysis, which is, in my view, a part of opinion mining.

Typically, opinion mining software is applied to the analysis of user's opinions uploaded by their authors to social networks, blogging, or microblogging systems. For more complete interpretation of these texts it is important to understand the structure of such networks and characterize their users and the interrelations between clusters of users. Social network analysis is a discipline of high importance for opinion mining.

Finally, the interpretation of mining results requires knowledge about the authors of the opinions, such as age or gender: say, do young people like iPhone 7 more than older people do? Do women prefer Clinton and men Trump? The task of identification of the authors' characteristics from a text is called author profiling. In addition, identifying documents written by the same author, or detecting the fact that documents uploaded by the same user in fact belong to different authors, is an important factor in tracking user preferences and their change over time, as well as in

detecting opinion spam and fake (paid) reviews, which are very numerous in Internet. Accordingly, I included in this issue a number of papers on author profiling and author identification.

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, listed at the end of this volume.

This special issue will be useful to researchers, students, and engineers working in sentiment and emotion analysis, social network analysis, natural language processing, computational linguistics, and related fields.

April 2016

Alexander Gelbukh,
Guest Editor

Table of Contents

Page

Let's Agree to Disagree: Measuring Agreement between Annotators for Opinion Mining Task **9**
Octavio Sánchez-Velázquez, Gerardo Sierra

ISAO: An Intelligent System of Opinions Analysis **21**
Sarra Zrigui, Rami Ayadi, Anis Zouaghi, Salah Zrigui

Computing Sentiment Polarity of Opinion WHY Type Question for Intention Mining of Questioners in Question Answering Systems **31**
Amit Mishra, S. K. Jain

Investigation of the Feature Selection Problem for Sentiment Analysis in Arabic Language **41**
Ahmed Nasser, Kıvanç Dinçer, Hayri Sever

A Proposed Lexicon-Based Sentiment Analysis Approach for the Vernacular Algerian Arabic **55**
M'hamed Mataoui, Omar Zelmati, Madiha Boumechache

A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis **71**
Heba M. Ismail, Saad Harous, Boumediene Belkhouche

Event Detection in Czech Twitter **85**
Václav Rajtmajer, Pavel Král

Blog Annotation: From Corpus Analysis to Automatic Tag Suggestion **95**
Ivan Garrido-Marquez, Jorge Garcia Flores, François Lévy, Adeline Nazarenko

A Lookup-Free Approach to Knowledge Extraction from News Feeds **107**
James Little, Chris Painter

Detecting Communities Using Link and Content Triangles **117**
Qiuling Yan, Baoli Li, Dongqing Yang

Author Profiling: Age Prediction Based on Advanced Bayesian Networks **129**
Seifeddine Mechti, Maher Jaoua, Rim Faiz, Heni Bouhamed, Lamia Hadrich Belguith

The Key Factors and Their Influence in Authorship Attribution **139**
Raheem Sarwar, Sarana Nutanong

An Analysis Framework for Hybrid Authorship Verification **151**
Seifeddine Mechti, Maher Jaoua, Rim Faiz, Lamia Hadrich Belguith

Let's Agree to Disagree: Measuring Agreement between Annotators for Opinion Mining Task

Octavio Sánchez-Velázquez and Gerardo Sierra

Universidad Nacional Autónoma de México, Grupo de ingeniería lingüística, Instituto de Ingeniería UNAM,
oct_sanc@unam.mx, gsierram@iingen.unam.mx

Abstract. There is a need to know up to what degree humans can agree when classifying a sentence as carrying some sentiment orientation. However, a little research has been done on assessing the agreement between annotators for the different opinion mining tasks. In this work we present an assessment of agreement between two human annotators. The task was to manually classify newspaper sentences into one of three classes. For assessing the level of agreement, Cohen's kappa coefficient was computed. Results show that annotators agree more for negative classes than for positive or neutral. We observed that annotators might agree up to a level of substantial agreement of 0.65 for the best case or 0.30 for the worst.

Keywords: inter-annotator agreement, opinion mining, corpus labelling

1 Introduction

Nowadays there are more user-generated content than ever before. This allows people to quickly share and find all sort of information. This information is split in two big kinds: objective information, which is all the information that we can measure or prove, and subjective information which comprises appraisals, thoughts and opinions [20, 10]. The computational study of the latter is known as opinion mining. There has been several approaches to achieve this. The main two have been machine learning classifiers, which require tagged examples as inputs [14, 1], and lexicon based systems, which require lexicons with each word classified with its polarity (negative, positive, neutral) [6, 17, 12].

In order to create systems capable of detecting and classifying opinions written in texts, we need human labelled corpus, either to evaluate, generate the models or train a system. When extracting opinions from product reviews, there is usually a score readily available. This is because pages that let make a review of a product usually have a system for ranking the product being reviewed, usually with a system of stars. There are some domains, as newspaper articles, where the label for the document or the sentence are lacking. For these kind of texts there is a need of documents labelled by human annotators. While usually people might agree on marking objective tokens from the reality, when dealing with subjective information, such as opinions, there are several variables that

depend on the experiences of the annotator. Some of these might be the domain expertise, the personal opinion on some domain or subject, point of view with respect some idea, ambiguity of language or ambiguity on interpretation [2, 11, 4].

Even though this is known, just some papers report the agreement of the annotators. There are several works assessing inter-annotator agreement in different tasks, such as image annotation [13], part-of-speech tagging [3], word sense disambiguation [19]. There are also work done on other areas, as biology [7] or medicine [8]. As far as we know, there are just few works on opinion annotation agreement. These works are not as conclusive as they could be.

In this paper, we present a work on inter-annotator agreement assessment and we try to show that this agreement is lower than in some other areas. We used Cohen's kappa to assess the degree of agreement of two annotators. To explain this, in section 2 we review some work done in particular on opinion mining labelling agreement. Then on section 3, we briefly describe the Cohen's kappa (κ) that was the measure we used to assess the agreement. On section 4 we describe the data used for assessing this agreement and describe the experiment. We discuss the results of these experiment on section 5. Finally, in section 6 we present our conclusions and future work.

2 Related Work

In [16], the authors compare the agreement between experts and non experts. They use the Amazon Mechanical Turk (AMT) to perform five different tasks. These were affect recognition, word similarity, recognizing textual entailment, temporal event recognition, and word sense disambiguation. What the assess is how much did AMT agreed with a gold standard annotated by experts. For the task of affect recognition (which is a similar one to opinion mining), they used headlines of newspapers. To assess the agreement they used Pearson correlation. They report an agreement of 0.576 between expert annotators and 0.417 between a Non-expert and an expert. However when they compared the correlation between an expert and the average classification of all other experts and non-experts, they got 0.603. For the other tasks, their correlations were about 0.96. Which means that it is harder to agree with other people respect affect recognition.

In [5], the authors present an annotation scheme for adding entity and event target annotations to the MPQA corpus [18]. While they describe the added features to the MPQA, they have a section dedicated to the agreement. For the annotations, they developed a manual and gave it to the annotators. After the annotator were trained, they tagged four documents. To asses the agreement, they computed the F-measure of the documents taking turns of documents annotated by annotator A and those of annotator B as gold standard. By doing this, they report an agreement of 0.82. To solve the disagreement, annotators could discuss the situation so, after doing this, they could agree.

In [15] the authors present a method for creating a dictionary and marking the words with emotions or polarity. They annotated the words by multiple annotators. They used six basic emotions to annotate the words. They analysed the tags given by multiple annotators using a weighted kappa adapted for multiple entries. They trained the annotators with a manual that instructed how to label the words. They obtained an average of 0.6452 for a lineal weighted kappa and an average of 0.7717 for a quadratic weighted kappa. Differently from our work, they tagged just words and the frequency of each word used in the context that represents one of the six basic emotions. Based on this they propose a a factor of probability of affective use (FPA).

3 Cohen's kappa

Cohen's kappa is a statistic which assesses the inter-annotator agreement for categorical items. Opposite to a simple percent agreement computation, this measure takes into account that some agreement might take place by chance. It is thought to measure how much two annotators, when each classify N items into C mutually exclusive categories, agree with each other.

If we build a square matrix where each column represent the instances sorted in a category by one annotator and each row represents the instances sorted in the category by the other annotator, we could have a confusion matrix. In this matrix, we should have a correspondence in categories in each row and column. This means that the diagonal would represent the coincidences while classifying the instances.

In order to get the value of kappa (κ), it is needed to know the observed agreement. The observed agreement is how much, of the total, did the annotators agree. This is, what is the sum of the diagonal of our confusion matrix over the total of instances:

$$P(o) = \frac{\sum_{i=1}^{|C|} C_{ii}}{N}$$

where C is the confusion matrix containing the categories that were classified and in each cell the total of instances that were classified into that category for each annotator; and N is the total of instances that were classified.

However, it is possible that some of the coincidences between the two annotators is given by chance. In order to rule out this possibility, the hypothetical agreement by chance should be computed. This is done by:

$$P(e) = \sum_{i=1}^{|C|} \frac{C_{i+}}{N} \times \frac{C_{+i}}{N}$$

where C is the confusion matrix, C_{i+} is the sum of all the counts in row i and C_{+i} is the sum of all the counts in column i, and N is the total of instances that were classified.

Thus, in order to know the real agreement between the two annotators, it is needed to subtract the probability of chance agreement from the observed agreement. Once we take this into account we have the Conhen's kappa, given by:

$$\kappa = \frac{P(o) - P(e)}{1 - P(e)}$$

where $P(o)$ is the observed agreement and $P(e)$ is the hypothetical probability of chance agreement.

One aspect that has to be observed, is that this kappa assumes its theoretical maximum value of 1 only when both observers distribute codes the same. So, when it is the case that corresponding row and column are not identical, we need to compute the maximum possible value and use that to interpret in a better way the agreement assessed. This is done by the next formula:

$$\kappa_{max} = \frac{P_{max} - P(e)}{1 - P(e)}$$

with:

$$P(e) = \sum_{i=1}^{|C|} \frac{C_{i+}}{N} \times \frac{C_{+i}}{N}, P_{max} = \sum_{i=1}^{|C|} \min\left(\frac{C_{i+}}{N}, \frac{C_{+i}}{N}\right)$$

where $P(e)$ is the same as before and $Pmax$ is the sum of the minimum number between the row and the correspondent column over the total of instances classified N.

This help us to interpret the obtained kappa as a proportion of the theoretical maximum agreement that annotators could have.

Nevertheless, interpreting the value of kappa is a difficult task for different factors like the distribution of the probabilities of the categories, and the number of categories itself. However, there are some guidelines that has appeared and are widely used, particularly in the areas of health sciences and humanities. Arguably the most used are those proposed by Landis and Koch [9]. This is shown in the table 1

Table 1. Interpretation of kappa value

Kappa Statistic	*Strength of agreement*
<0.0	Poor
0.00 - 0.2	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Almost perfect

In this table, it is possible to see that according to the values of kappa, the strength of the agreement varies. We are going to use these guidelines and the maximum kappa in order to help in the interpretation of our values. We are going to use a division of the κ over the κ_{max} in order to obtain the proportion of agreement normalised by the maximum expected value.

4 Experimental set-up

In this section we explain the data we used to perform the inter-annotator agreement assessment and present the results of different evaluations.

We used a corpus of newspapers from Mexico. The news were dated between 1st of January, 2014, and 31st of October of 2014. All the news were manually selected as being about finances. The total of news articles is 300. From this we randomly selected 10 news.

From those selected news, we split each one into sentences. By doing this, we obtained 151 different sentences.

Then, we asked two annotators about the same age (21 yo), same gender, and same level of studies (last year of bachelor in language and arts), to label each sentence in each article as positive, negative or neutral, with no further explanation. The annotators did not know that another annotator was doing the same task. Finally we compared the sentences that both annotators labelled and compare if they agreed in the selected category or not. With those labels we created a confusion matrix containing in columns the distribution of categories given by annotator 1 and in rows those given by annotator 2.

After doing this we obtained the following results.

For the positive category, we got:

Table 2. Sentences labelled as positive

		Annotator 1		
		Yes	No	Total
Annotator 2	Yes	**38**	7	45
	No	45	**61**	106
	Total	83	68	**151**

As it can be seen in table 2, annotator 1 and annotator 2 agreed in 99 sentences. From this, they agreed that 38 were positive and 61 were not positive, thus they could be neutral or negative. It is also observable that while Annotator 1 labelled a total of 83 sentences as positive and 68 as not positive, Annotator 2 thought that just 45 sentences were positive, while 106 were not positive. So it is feasible to think that the agreement is as low as $\frac{38+65}{151} = .6556$. However, this is just the observed agreement.

For this particular case we have that $P(e) = .4799$. So in order to know the real agreement between the two annotators, it is needed to subtract the

probability of chance agreement from the observed agreement. For this particular case we know that the agreement without chance is $\kappa = 0.3378$. This would be interpreted as having a **fair** strength of agreement, according with table 1.

However, if we compute the maximum kappa for this category, we get that $\kappa_{max} = 0.5161$. so, if we take this into account, the proportional kappa, normalised by its maximum, would be of $\kappa = 0.6545$. So in this case we could see it as having a **substantial** strength of agreement.

When we compared the sentence labelled as neutral, we found the distribution expressed on table 3.

Table 3. Sentences labelled as neutral

		Annotator 1		
		Yes	No	Total
Annotator 2	Yes	**34**	46	80
	No	8	**63**	71
	Total	42	109	**151**

In table 3 we observe that agreement is similar to that observed in table 2, that is, both annotators agreed that there were 34 neutral sentences and 63 non-neutral sentences. However, when looking at the totals, it is noticeable that while Annotator 2 thought there were 80 neutral sentences in total, Annotator 1 thought there were just 42. This mean that Annotator 2 thought there were 71 sentences that were not neutral, thus either positive or negative, while Annotator 1 thought there were 109 sentences being not neutral.

Because of this, the observed agreement is similar to, but a little lower than, that observed in positive. This is $P(o) = 0.6424$. Also the probability of chance agreement is a little higher than that for positive. For the neutral class, we computed $P(e) = 0.4868$. So, when we take into account the two facts, a little lower $P(o)$ and a little higher $P(e)$, it is normal to expect that κ coefficient would go lower, as it is the case. In this particular class, we computed a $\kappa = 0.3032$. This means that according to the interpretation table, we would evaluate this as having a **fair** strength of agreement.

As expected, the maximum kappa for this category is similar to that computed for positive. In this particular case we have that $\kappa_{max} = 0.5097$. So taking this into account, we get that the proportional kappa for the maximum expected agreement would be $\kappa = 0.5949$. This is considered to have a **moderate** strength of agreement between the annotators.

After looking at the sentences labelled as negative, we observed an increased agreement between the annotators. The table 4 shows how they distributed the sentences.

It is noticeable that agreement is much higher in this case. It can be observed, also, that differently from the previous cases, in this one, the matrix is symmetric. In this case, the annotators strongly agreed on what was not a negative sentence.

Table 4. Sentences labelled as negative

		Annotator 1		
		Yes	No	Total
Annotator 2	Yes	**15**	11	26
	No	11	**114**	125
	Total	26	125	**151**

Also both annotators agreed that there was just 26 negative sentences. This left 125 as non-negative sentences.

Thus, the observed agreement was $P(o) = 0.8543$. This is higher than the positive and neutral. This is because both annotators labelled many sentences as being non-negative.

We computed the hypothetical agreement by chance as being $P(e) = 0.7149$. This is also higher than in the other two cases. When these two factors are taken into account, the agreement coefficient is $\kappa = 0.4889$. This is notoriously higher than in the other two cases.

This means that it there will be more agreement on classifying sentences as negative or non-negative than in positive or non-positive and in neutral or non-neutral.

This is, annotator agreed on classifying 114 sentences as non-negative, and they both classified 125 as non-negative. However they didn't agreed much on classifying those sentences as positive or neutral. While annotator 1 considered 83 as being positive and 42 as being neutral (table 2, table 3), annotator 2 distributed their non-negatives almost flipped: 45 as being positive and 80 as being neutral.

For this particular case, the maximum possible kappa is of $\kappa_{max} = 1$. This is because the table is completely symmetrical.

This is interesting because it shows that it is much harder to agree on positiveness or neutrality of a message than it is to agree on their negativity.

In table 5 we can observe a summary of the results of the three tables.

Table 5. Kappa by category and it's interpretation

Category	*Observed Kappa*	*Interpretation*	*Maximum Kappa*	*Proportional*	*Interpretation*
Positive	0.3378	fair	0.5161	0.6545	substantial
Neutral	0.3032	fair	0.5097	0.5949	moderate
Negative	0.4889	moderate	1	0.4889	moderate

When we take into account the total distribution of the labelled sentences, we observe more clearly how the annotators classified each sentence. The next table summarises this process.

We can observe in a more detailed fashion how the non-class were distributed by each of the annotators.

Table 6. Sentence distributed by the three classes

		Annotator 1			
		Pos	Neu	Neg	Total
Annotator 2	Pos	**38**	6	1	45
	Neu	36	**34**	10	80
	Neg	9	2	**15**	26
	Total	83	42	26	**151**

As we discussed earlier when talking about the positive labels, both annotators agreed on 38 sentences being positive. However, annotator 1 also marked other 45 sentences as positive. From those other 45 sentences, annotator 2 thought 36 were neutral and 9 negative. On the other hand, annotator 2 just added 7 more sentences than the 38 they agreed as positive. From these sentences, annotator 1 labelled 6 as neutral and 1 as negative. In total, annotator 1 labelled 83 sentences as positive while annotator 2 just labelled 45.

For the neutral case, we find a similar scenario, nevertheless it is flipped over. In this category, both annotators agreed on labelling 34 sentences as neutral. On the disagreed sentences, annotator 1 labelled just 8 more as neutral; from these, annotator 2 thought that 6 were positive and two negatives. Contrastingly, annotator 2 foresaw 46 more sentences as being neutral; annotator 1 considered 36 of these as being positive and 10 as being negative. In total, annotator 1 labelled 42 sentences as neutral while annotator 2 almost doubled it and labelled 80 as neutral.

At labelling the negative classes, we find a little different picture. In this case, as stated, both annotators agreed as labelling 15 as negative. Also both labelled, in total, 26 as negative, but from the disagreed 11 of the annotator 1, annotator 2 views 1 as positive and 10 as neutral. On the other hand, from those 11 that annotator 2 esteemed negative and disagreed with annotator 1, the latter classified 9 as positive and two as negative.

The observed agreement for the three classes together would be $P(o) = 0.5762$. This does not seem as high as one might expect. Even though the chance of agreement has not being subtracted from that number. This chance is $P(e) = 0.3408$. Therefore we might assess the Cohen's correlation coefficient being as $\kappa = 0.3570$. This is, according with Landis and Koch [9], **fair** agreement.

Nevertheless, it is important to take into account that kappa assumes its theoretical maximum value of 1 only when both observers distribute codes the same. So, when it is the case that corresponding row and column are not identical, we need to compute the maximum possible value and use that maximum to interpret in a better way the agreement assessed.

We get that maximum kappa is $\kappa_{max} = 0.6182$. This mean that if we use this maximum as a help to interpret the actual kappa obtained, we could think that the value is around $\kappa = 0.5775$. This falls, according to Landis and Koch, in the lower rank of a substantial or higher rank of moderate agreement. Another interpretation that is possible to make based on the k_{max}, is that annotators

will never agree in a higher degree. This mean that best agreement would be a moderate agreement.

5 Discussion

In this section we will discuss the results obtained during the experiment and will try to interpret them.

Based on the observations made on this experiment, we might notice several things. The most noticeable is that, when labelling opinions in the news, annotators without specific training will not agree in a high degree, specially when deciding if a sentence is positive or neutral.

We also noticed that the peak of agreement was achieved when deciding whether a sentence is negative or not. For this class, both annotators agreed that there were few negative sentences. Another interesting observation, is that even though both annotators agreed on tagging 26 sentences as negative, there were some sentences classified as positive by one annotator and negative by the other. This proofs that people will, at least sometimes, strongly disagree on the polarity of a sentence.

Another worthy observation is that while one annotator had a strong inclination for labelling sentences as positive (83), the other thought that many sentences were neutral (80). If we take into account that annotators are the same gender, same age and has same level of education, we might think that these differences are explained for their own experiences. This means that people, therefore annotators, will think of a sentence as being positive or neutral (or even negative) depending on their own perspectives and expectations of how the newspaper text should deal with the topic that it is about.

6 Conclusion and future work

From this experiment we can conclude that any classification system should take into account that agreement among annotators is more or less small. Because of this, it is possible to think that, under these conditions, any opinion mining system developed with this kind of corpus, will always be biased. This might be either a model bias or a learning data bias (if the system is non-supervised) or by the annotators that were used for labelling training data (in the case of supervised learning systems).

As future work we think it is important to compare these results with more data and assess agreement with other inter-annotator agreement metrics. We also think that it would be interesting to compare if the agreement is more or less the same when changing the input data, this is, with films or product reviews, with political speech, etc. Another thing we want to do in a future, is to compare the results after giving clear instructions of what to consider negative, neutral and positive to the annotators.

Acknowledgements

This work was done with the support given by the project IN403016 DGAPA-PAPIIT. The authors also want to thank the Consejo Nacional de Ciencia y Tecnología (CoNaCyT) for the support given through grant 387405 and project CB2012/178248.

References

1. Agarwal, B., Mittal, N.: Machine learning approach for sentiment analysis. In: Prominent Feature Extraction for Sentiment Analysis, pp. 21–45. Springer (2016)
2. Bermingham, A., Smeaton, A.F.: A study of inter-annotator agreement for opinion retrieval. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 784–785. ACM (2009)
3. Brants, T.: Inter-annotator agreement for a german newspaper corpus. In: LREC (2000)
4. da Cunha, I., Torres-Moreno, J.M., Sierra, G.: On the development of the rst spanish treebank. In: Proceedings of the 5th Linguistic Annotation Workshop. pp. 1–10. Association for Computational Linguistics (2011)
5. Deng, L., Wiebe, J.: Mpqa 3.0: An entity/event-level sentiment corpus. In: Conference of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies (2015)
6. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. vol. 6, pp. 417–422. Citeseer (2006)
7. Kaewphan, S., Van Landeghem, S., Ohta, T., Van de Peer, Y., Ginter, F., Pyysalo, S.: Cell line name recognition in support of the identification of synthetic lethality in cancer from text. Bioinformatics 32(2), 276–282 (2016)
8. Lacson, R., Raja, A.S., Osterbur, D., Ip, I., Schneider, L., Bain, P., Mita, C., Whelan, J., Silveira, P., Dement, D., et al.: Assessing strength of evidence of appropriate use criteria for diagnostic imaging examinations. Journal of the American Medical Informatics Association p. ocv194 (2016)
9. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. biometrics pp. 159–174 (1977)
10. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining text data, pp. 415–463. Springer (2012)
11. Molina, A., SanJuan, E., Torres-Moreno, J.M.: A turing test to evaluate a complex summarization task. In: Information Access Evaluation. Multilinguality, Multimodality, and Visualization, pp. 75–80. Springer (2013)
12. Nielsen, F.Å.: A new anew: Evaluation of a word list for sentiment analysis in microblogs. arXiv preprint arXiv:1103.2903 (2011)
13. Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: Proceedings of the international conference on Multimedia information retrieval. pp. 557–566. ACM (2010)
14. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)

15. Rangel, I.D., Guerra, S.S., Sidorov, G.: Creación y evaluación de un diccionario marcado con emociones y ponderado para el español. Onomazein 29(1), 31–46 (2014)
16. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of the conference on empirical methods in natural language processing. pp. 254–263. Association for Computational Linguistics (2008)
17. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61(12), 2544–2558 (2010)
18. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language resources and evaluation 39(2-3), 165–210 (2005)
19. Yong, C., Foo, S.K.: A case study on inter-annotator agreement for word sense disambiguation. In: Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources (SIGLEX99) (1999)
20. Zhang, L., Liu, B.: Aspect and entity extraction for opinion mining. In: Data mining and knowledge discovery for big data, pp. 1–40. Springer (2014)

ISAO: An Intelligent System of Opinions Analysis

Sarra Zrigui[1], Rami Ayadi[2], Anis Zouaghi[3], Salah Zrigui[4]

[1,3] ISSAT Sousse, Tunisia, LATICE Laboratory
sara.zrigui@gmail.com, Anis.zouaghi@gmail.com,
[2]LaTICE laboratory, University of Sfax, Sfax, Tunisia
ayadi.rami@planet.tn
[4]École Nationale Supérieure d'Informatique et de Mathématiques
Grenoble, France
salahzrigui@gmail.com

Abstract. Today, the need to automatically process opinions is strongly felt. It is in this context that we situate this work whose objective is to contribute to the achievement of opinions analysis system, enabling a binary classification on a set of textual data. For this, we studied and evaluated several methods, Support Vector Machines (SVM) and Naïve Bayes (NB), on a corpus composed of 500 journals films. These models have not been satisfactory. To improve the results we have introduced a pre-treatment phase or standardization corpus before classification; this phase has improved the quality of the classification.

Keywords: opinions analysis, Arabic, classification, SVM, NB.

1 Introduction

Nowadays, the amount of information generated by users is increasing very rapidly. Whether on forums, blogs, e-commerce sites or social media sites, users continue to share their knowledge and their views on products, ideas, events, etc. This large amount of opinions can influence the way to perceive brands, people, organizations and events, which can motivate the masses to action. Hence the need to create systems for classification and analysis of opinions was born. To meet this need, many researches have emerged. They come from different areas: data mining, decision support, knowledge modeling, natural language processing etc.

The social internet such as social networks, forums, blogs, e-shopping sites, etc. detonated the number of texts expressing opinions. Millions of messages appear every day in social networks such as Twitter and Facebook for example.

Increasingly such sites are used by users to post their opinions about products and services they use, or express their political and religious views. These websites have become a very valuable source of opinions and feelings of people. This social data can then be used for:

- Marketing by analyzing the views of users about a product or brand.
- Social studies analyzing the societal trends. etc.

The need to automatically process opinions is therefore strongly felt. It is in this context that this work is inscribed, and whose objective is:

- To contribute to the realization of an opinions analysis system, enabling a binary classification on a set of textual data.
- To study and evaluate the effectiveness of support vector machine (Support Vector Machines (SVM) in English) and the naive Bayesian model (Naïve Bayes (NB)) to perform this task.
- To consider and test the impact of a number of pretreatments on the analysis of views and system performance.

This paper consists of five sections; in the second section we'll describe the Arabic language and its complexity. In the third, we present some related work. In the fourth section we present our contribution. Finally we end this paper with a conclusion.

2 Problems of Opinion Analysis in Arabic

2.1 Arabic Language

Arabic is the fifth most used language in the world. It is the mother tongue of over 200 million people and more than 450 million speakers [1]. The Arabic language is considered by Internet World Stats [2] as the language with the fastest growth rate in terms of internet users in the last eleven years. The Arabic language has three forms; namely Classical Arabic (CA), Modern Standard Arabic (MSA), and the Arabic dialect (DA). CA is one used in conventional historical texts, the MSA is the language used by the media and in official speeches, and finally the DA consists mainly dialects spoken and has no written standards [3]. The Arabic alphabet consists of 28 letters, unlike Latin alphabets, the orientation of the Arabic writing is from right to left. Unlike the English language, for example, the notion of upper and lower case does not exist in Arabic.

2.2 Complexity of Automatic Processing of the Arabic Language

Arabic is a difficult language to automatically deal with for several reasons, among which we can mention [4]:

- The presence of diacritics makes it a less ambiguous and more phonetic language, but unfortunately the majority of texts are not vowelized.
- Certain combinations of characters can be written in different ways.
- A very complex morphology compared to the English language.
- Synonyms are widespread. The Arabic language is a highly inflected language and derivational.
- Lack of publicly available Arabic corpora.
- Lack of Arabic digital content.

Meaning of words. A word can have more than one meaning depending on the context in which it is used. A word can have more than one lexical category (noun, verb, adjective, etc.) in different contexts. The following figure shows an example of this ambiguity.

Synonyms. There are many words that are considered synonymous. Given a corpus, researchers can use the tools of morphological analysis to find synonyms of a word, the frequency of each word of these synonyms and if one of them is more common

Form of words depending on its mode of use. The form of some Arabic words can change according to their case modes [16] (nominative, accusative or genitive). For example, the plural of a word (مسافر) meaning (traveler) may be in the form (مسافرون) in the case of nominative (مرفوعة) and shape (مسافرين) in the case of the accusative / genitive (مجرورة / منصوبة).

Morphological characteristics. An Arabic word consists of a root, more affixes and clitics. The stem consists of a consonant root (صحيح جذر) and a pattern of morpheme (معني ذات كلمة اصغر). Affixes include time markers, sex and / or numbers (اعرابية حركات). Clitics include some prepositions (الجر حروف), conjunctions (العطف حروف) determinants (داتمحد), possessive pronouns (الملكية ضمائر) and pronouns (ضمائر) [17]. The Stemming process reduces the number of features extracted from a corpus by converting the words to their stems. There is another approach to the reduction of the morphology which simply removes affixes and does not convert the word to its stem. This approach is called Light Stemming.

3 Related Work on Arabic

The work done in the context of Arabic is limited to the work performed on the application of different classification techniques and also of the work on the pre-treatment applied on the text before the classification process.

In this paper [5], Al-Kabir, address the issue of the effect of the Stemming the classification of Arabic text documents. It applies the text classification for text documents using Stemming in the pretreatment steps. The results showed that the support vector machine (SVM) classifier has reached the precision of the highest classification using the two test modes with 87.79% and 88.54%.

The main objective of this study [6] is to measure the accuracy for each classifier to determine which is more accurate for the Arabic text classification based on function words. Classifiers are studied Support Vector Machine (SVM) with sequential optimization Minimal (SMO), Naive Bayes (NB), and J48. The results show that the use of SMO provides the highest accuracy and lowest error rate, and that the time needed to build the SMO model is the smallest time.

There are several studies that compare the performance of different classification algorithms on Arabic text. In [7], Alsaleem studied the performance of methods, Bayesian Naïve (NB) algorithm and Support Vector Machine (SVM), on different

sets of Arabic data. The experimental results against various Arab text data sets show that SVM algorithm outperforms the NB with regard to all measures.

In [8], the author compared the performance of KNN and SVM. This study showed that both have superior performance, and SVM improved accuracy and time. In [9], the author have applied KNN and NB on the text in Arabic and have concluded that KNN has better performance than NB, they also concluded that the selection of features and the size of the overall training and value K affected the performance of the classification.

EL-HALEES [10] compared six well-known classifiers applied to the Arabic text; ANN, SVM, NB, KNN, maximum entropy and the decision tree. It showed that NB and SVM classifiers are the best in terms of F-measure with 91% and 88% respectively. In [11], Al-Khorsheed thubaity studied a variety of text classification techniques; SVM, Knn, NB using the same data sets that belong to a wide range of categories.

4 Experiment and Evaluation

In this part we will present the experimental study. We will start by showing the tools that have contributed to this work and the approach.

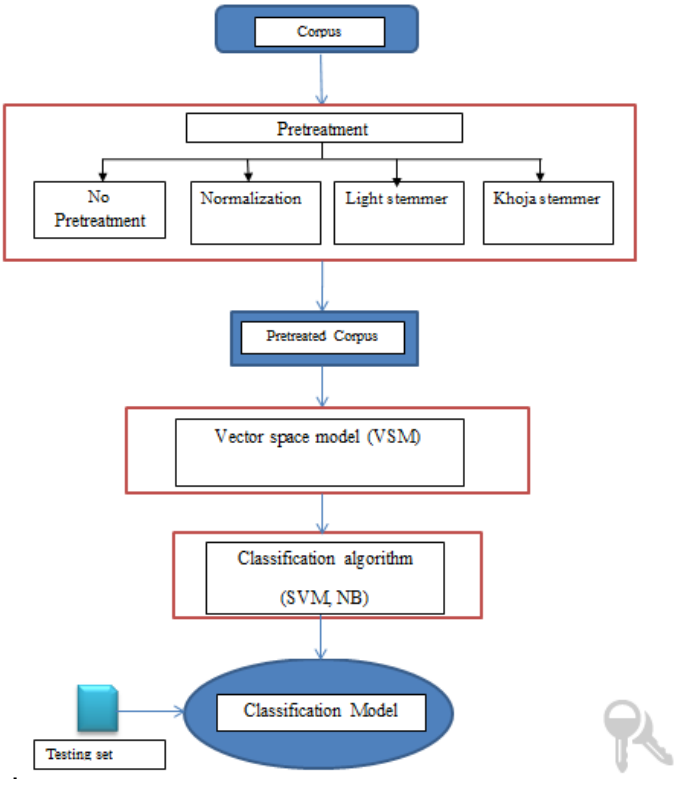


Fig. 1 Classification process

4.1 Preprocessing

Generally natural language texts cannot be directly analyzed (interpreted for example by a classifier or by classification algorithms)[18]. Whatever type of data, it is necessary to pretreat the raw data in order to then treat it with unified processes and not with a multitude of processes adapted to all possible cases [12].

These pretreatments are done to standardize the different ways of writing the same word to get correct results, to correct obvious spelling or typographical inconsistencies and to clarify certain lexical information implicitly expressed in the texts and some missing information to the using external resources.

It is necessary to use these treatments according to the final objective in order clarification or maximize the number of operations performed. Many of these treatments are specific to the used language (it is not the same type of pretreatment to documents written in English and French or Arabic).

At the simplest level, pretreatment is to index and count all the words found in the entry documents to calculate a table of documents and words, ie, a frequency matrix that lists the number of appearances of each word. This basic process can be refined to exclude some common words such as "الذي" and "ذلك" (empty word) and combine different grammatical forms of the same words such as "مسافرين", "مسافرون", "مسافر" etc.

In this step, the Arabic texts are transferred to a format suitable for Stemming process. Generally, punctuation and special characters are removed. This is followed by the application of certain linguistic processing. Some of the most popular treatments are [13][15]:

Each document in the set of data is processed to Arabic
Delete all numbers and punctuation marks.
Remove all vowels except (. (ا ل ش د ة
Duplicate all the letters containing the symbols (shadda).
Converting "أ" ,"إ" and "آ" to "ا".
Convert "ى" to "ي" and "ة" to "." ه
All non-Arabic words are filtered.
Arabic empty words are removed.
Apply a Stemming Algorithm.

In the sequel, we present the algorithms we used for the pretreatment of the data.

Removing unnecessary characters. Removing unnecessary characters is performed by following the steps of the following algorithm:

Algorithm 1 : Removing useless characters

d a document of *N* word *w* $d=(w_1,\ldots,w_n)$
Input: text
Output: text without useless characters
For each word w_i of d do

1. Delete all punctuation marks
2. Delete all latin numbers and characters
3. Delete all abbreviations and isolated letters

- Punctuation: This step will remove any sequence of punctuation characters delimited by letters or spaces such as the comma and semicolon ... etc. In Arabic texts, some characters are written from right to left as the question mark "?"And the comma", ", this orientation is also taken into account during processing.
- The numbers and Latin characters: Here all character sequences located between two spaces containing Arabic numerals "1 ... .9" or Latin numbers or Roman «1 ... .9 ... I. IX" are eliminated, and we also eliminate the Latin letters "A ... Z, a ... z".
- The abbreviations and single letters: The abbreviations of words, such as: ت for "تاريخ = date" م to ميلادي ,ص for "صفحة = page" ج for "جواب = answer," س for "سؤال = question" . or coordination as ك ,ل ,ف ,و ,ب (bi-, wa, fa-, li-, ka ...) scored as isolated forms next to the numbers (e.g. 5 ب 32) or the mathematical formulas as (3 + ع س =).

Removal of empty words. The empty words are the words that frequently occur in most documents in a given collection without significant semantic relation to the context in which they exist. They will be removed from the text because their presence or absence does not provide useful information on the meaning of the text. The figure 2 contains a partial list of empty words. In Arabic, the list of stop words can include the punctuation marks (!? ...), Pronouns (... هم الذي التي هي هو), adverbs (... أمام تحت فوق خلف), days of the week (... الاثنين الثلاثاء الاربعاء) months of the year (... اكتوبر نوفمبر سبتمبر). There is no definitive list of empty words used in all tools. They are also tools that use no stoplist. Some tools specifically prevents their use to support the search phrase.

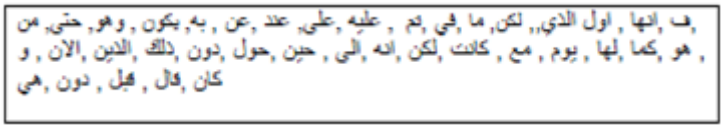


Fig. 2 partial list of empty words

Morphological processing. The morphological character normalization was achieved by following a number of rules. These rules are defined by:

Algorithm 2 : Morphological processing

Input: text

Output: text after morphological traitments

For each word w_i of *d* do

1. Convert all « إ » ,« أ » and « آ » to « ا »
2. Convert all “ى” to “ي” and “ة” to “ه”
3. Delete the character ‘—’
4. Delete all vocalization signs : «ً,ٍ,ٌ,ِ,ُ,َ»
5. Duplicate characters with « ّ »

- The first step of this algorithm is to normalize "Alif and Hamza", it is to convert el "أ" ,"إ" and "آ" to "ا". The reason for this conversion is that all forms of Hamza are

represented in dictionaries "ا", as most of the texts neglect adding Hamza El Alif and often ill people spell the different forms of aleph.

- The second step is standardizing ي and ة "Yâ' and el tâ marbouta". The 'Hamza' character adds to the confusion, whether it is at the end of the word, between ي (Letter yâ final) and ى (or 'alif maqsura): The word نادي, Nadi, "club", can be noted نادى, (read as nâdâ, "invite ").
- The third step is to delete the character '—' (kashida) because typesetters make frequent use of the '—' character (called kashida). These characters lengthen the line in the middle of words which allows for a clearer readability and reduce white space on a line or justified for purely calligraphic reasons. This character, not part of the Arabic alphabet, is often a source of confusion for the treatment of texts.
- In the last two steps we remove the signs of vocalization, all signs of vocalization ،ً, ٍ, ٌ, َ, ُ, are eliminated except for "" where we duplicate the character that contains it.

4.2 Results

Data set. The corpus used is the OCA corpus (Opinion Corpus for Arabic), developed by Rushdi Saleh and al [4].This corpus consists of 500 reviews for movies collected from various Arabic website and blog, classified respectively 250 250 positive reviews and negative reviews.

Tools. Several resources available on the web are ready to be used. We used a different combination of several tools developed in other research.

For pretreatment: Regarding the stemmers used the light stemmer Arabic and Arabic stemmer, the code is written in Java and published on the Internet developed by Mr. Saad [4].

For representation: to model the corpus, we also used the famous weka [14].

For classification: We used weka that implements a large collection of machine learning algorithms for data mining tasks.

Experiments. For the validation of our work, first preprocessing is performed to normalize each document in the corpus.

We made several experiments were prepared four combinations ready data polarity detection phase:

- The corpus without pretreatment.
- The corpus + Application of Standard
- The corpus + Application of arabic light stemmer
- The corpus + Application of Khoja stemmer
- Then we made a representation in words vector with the Weka tool.
- Then the classification, also using weka, is performed using these two classifiers SVM, NB.

The performances are evaluated using three metrics (precision, recall and F-measure).

In our case, 80% of the data is used for the training set and 20% is for the test. The following sections show the results obtained using this method.

We found that the best result was obtained using the method NB with a percentage of 82% classified document correctly, and that by the second experiment, the corpus + application standardization.

We still noticed that the application of the arabic light stemmer and Khoja stemmer degraded performance even compared to the result obtained using the corpus in its raw state. The following figure shows this clearly.

For the SVM classifier, we note that the best result was obtained with the corpus in the raw materials with 87% classified correctly. We also note that the application of arabic light stemmer and Khoja stemmer degraded performance. The results are displayed in the following figure.

The comparison between the results obtained by NB and SVM, shown in the table below, shows the superiority of NB classifier in all tests. The figure illustrates well the results:

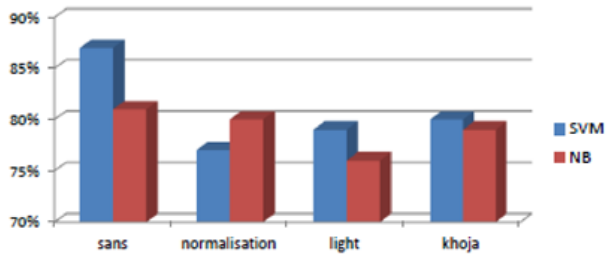


Fig. 3 NB, SVM comparaison (percentage split)

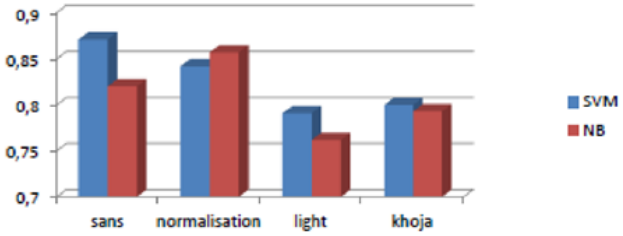


Fig. 4 NB, SVM precision comparaison (percentage split)

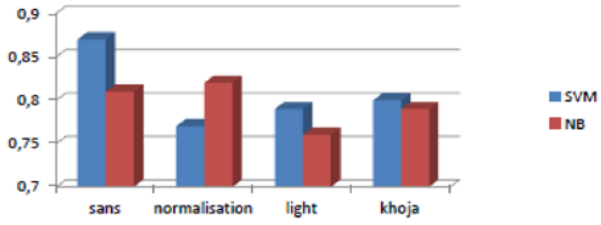


Fig. 5 NB, SVM recall comparaison (percentage split)

The application of light and stemming Khoja also degraded results. Here SVM was most effective with most pre-treatments, but it reached 0.871 precision and recall of 0.87 without any pretreatment with the estimation method of reliability "Percentage split."

The application of two classification algorithms, Naive Bayes (NB) and Support Vector Machines (SVM), on an Arabic corpus showed the superiority of the first over the second classifier. In the experiments we performed, we found that the naive Bayesian classifier gave the most successful results most of the time.

The results show the impact of the pretreatment phase and the application of different techniques with respect to a data set. The best results were achieved after the normalization of the corpus. The use of stemming and light stemming and Khoja stemming, degraded performance in most cases.

Despite its simplicity and the fact that the hypothesis of conditional independence obviously does not hold in real world situations, the NB classifier still tends to give effective results. Secondly, SVM has been shown to be very effective in the categorization of traditional text, usually surpassing NB.

5 Conclusion

This paper was primarily an exploration of the field of opinion analysis in Arabic. Throughout this work we encountered several challenges.

Among these challenges, we report the search for a reliable corpus tested by other similar research and explore a new area of research, namely natural language processing and especially Arabic.

We performed a binary classification (positive or negative) on 500 reviews of films. The two classifiers are used SVM and NB. This allowed us to compare the performance of two classifiers that are widely used in the classification field.

The experimental results showed the superiority of the classifier NB in most tests. The best result was obtained by performing a normalization of the corpus before the classification: this is our main contribution.

We also noted that the application of light stemming and Khoja stemming downgraded the results of our analysis of opinion.

References

1. About The Arabic language. World langages and culture. [En ligne] [Citation : 12 12 2015.] http://www.vistawide.com/arabic/arabic.htm.
2. About The Arabic language. (s.d.). Consulté le 12 12, 2015, sur World langages and culture: http://www.vistawide.com/arabic/arabic.htm
3. Mountassir, A., Benbrahim, H., & Berrada, I. (2012). SENTIMENT CLASSIFICATION ON ARABIC CORPORA: PRELIMINARY RESULTS OF A CROSSSTUDY. 3 e Séminaire de Veille Stratégique, Scientifique et Technologique (VSST'12).
4. Saad, M. K., & Ashour, W. (2010, November). Osac: Open source arabic corpora. In 6th ArchEng Int. Symposiums, EEECS (Vol. 10).

5. Al-Kabi, M., Al-Shawakfa, E., & Alsmadi, I. (2013). The Effect of Stemming on Arabic Text Classification: An Empirical Study. Information Retrieval Methods for Multidisciplinary Applications, 207.
6. Al-Shargabi, B., Al-Romimah, W., & Olayah, F. (2011, April). A comparative study for Arabic text classification algorithms based on stop words elimination. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications* (p. 11). ACM.
7. Alsaleem, S. (2011). Automated Arabic Text Categorization Using SVM and NB. Int. Arab J. e-Technol., 2(2), 124-128.
8. Hmeidi, I., Hawashin, B., & El-Qawasmeh, E. (2008). Performance of KNN and SVM classifiers on full word Arabic articles. *Advanced Engineering Informatics*, *22*(1), 106-111.
9. Moh'd Mesleh, A. (2011). Feature sub-set selection metrics for Arabic text classification. Pattern Recognition Letters, 32(14), 1922-1929.
10. El-Halees, A. (2008). A comparative study on Arabic text classification. Egyptian Computer Science Journal, 30(2).
11. Khorsheed, M. S., & Al-Thubaity, A. O. (2013). Comparative evaluation of text classification techniques using a large diverse Arabic dataset. Language resources and evaluation, 47(2), 513-538.
12. Heitz, T. (2006). Modélisation du prétraitement des textes. In JADT'06 (International Conference on Statistical Analysis of Textual Data) (Vol. 1, pp. 499-506).
13. 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.
14. Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.
15. Mounir Zrigui, Rami Ayadi, Mourad Mars, Mohsen Maraoui: Arabic Text Classification Framework Based on Latent Dirichlet Allocation. CIT 20(2): 125-140 (2012)
16. Mbarek Charhad, Mounir Zrigui, Georges Quénot : Une approche conceptuelle pour la modélisation et la structuration sémantique des documents vidéos, SETIT-3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, TUNISIA,(2005)
17. Mohamed Achraf Ben Mohamed, Sarra Zrigui, Anis Zouaghi, Mounir Zrigui: N-scheme model: An approach towards reducing Arabic language sparseness. ICTA 2015: 1-5
18. Mohamed Achraf Ben Mohamed, Souheyl Mallat, Mohamed Amine Nahdi, Mounir Zrigui: Exploring the potential of schemes in building NLP tools for arabic language. Int. Arab J. Inf. Technol. 12(6): 566-573 (2015)

Computing Sentiment Polarity of Opinion WHY Type Question for Intention Mining of Questioners in Question Answering Systems

Amit Mishra,[1] S. K. Jain[2]

[1]Computer Science & Engg Deptt, PCE, Jaipur, Rajasthan, India- 302033
[2]Computer Science & Engg Deptt, NIT Kurukshetra, Haryana, India- 136112

{amit.mishra@poornima.org, skj_nith@yahoo.com}

Abstract. Opinion question answering systems (OQAS) search for answers from public opinions available on social web. WHYquestions asked in OQAS expect answers to incorporate reasons and explanations for the questioners' sentiments expressed in the questions. Sentiment analysis has been recently used for determining sentiment polarity of WHY-questions so as to find the intention of users with which he is looking for getting information related to products. In our recent research[14, 15], we address complex comparative WHYtypes questions and propose an approach to perform sentiment analysis of the questioners. For example, the question, "I need mobile with good camera and nice sound quality. WHY should I go for buying Nokia over Samsung?" we determine the main focused product (Nokia) with respect to questioner's perspectives who shows positive intention for buying mobile. The work does not deal with questions that have mixed emotions like WHY Dells are ok, HPs aren't that good, but Macs are Fantastic. Moreover, the work does not perform feature specific (camera and sound quality) sentiment analysis of questioners. In this paper, we perform the feature based sentiment analysis of questioners. We also address complex questions that have mixed emotions towards different products. We examine semantic structures of questions and propose an approach for sentiment analysis of questioners on product review sites. We finally conduct experiments which obtain better results as compared to existing baseline systems.

Keywords: Question Answering; Information retrieval; natural language processing; natural language understanding and reasoning.

1 Introduction

Opinion Question Answering System (OQAS) retrieves answers from user generated data on web. Research on the complex questions such as 'WHY' has been very constrained [14, 15, 18, 19].

Sentiment analysis has been recently used for determining sentiment polarity of WHY-questions so as to find the intention of users with which he is looking for getting information related to products [1, 2, 4, 5, 6]. Research related to WHY-opinion ques-

tions consider simple WHY-questions expressed in single sentence [1, 3, 4, 25, 26]. TAC 2008 competition incorporated only simple WHY type questions like, "WHY did people like Megan" [25, 26].

In our recent work [15], we propose a method for identifying the key opinionated span within WHY questions asked on product review sites. We make use of a discourse parser [9]to fragment questions into different text segments for effective opinion mining. Subsequently, the polarity score of the most importanttext segment is computed using knowledge based approach with the help of semantic role labeler [13].

For example, I need mobile with good sound quality and nice looks. WHY should one feel happy after buying x?Our method traces 'x' as main focused product with the intention (positive) of questioner with respect to 'x'. There are limitations in the work. The work does not perform feature specific (sound quality, and looks) sentiment analysis. It is common that a questioner may have positive intentions for some features and negative intentions for other features of a product.

Secondly, the approach could not find main focused product in questions like, "If I need a great mobile that could handle basic computing needs. WHY should I go for Samsung over Nokia?", "WHY Dells are ok, HPs aren't that good, but Macs are Fantastic", "WHY there are more favorable comments about Nokia than Micromax", "WHY people are incredibly more dissatisfied with battery backup of Nokia in comparison to Samsung" etc.

In this paper, we propose a method to perform feature based sentiment analysis of questioners from the questions. We also determine the main focused product with respect to questioner's perspectives. In this regard, we perform semantic analysis of WHY Questions through a parser Enju[20]. We split the question into different group based on features of a product. Further, we identify feature related expressions in WHY type questions and compute the sentiment polarity of WHY type questions based on different features described in questions.

In summary our contribution is as follows:

1. We make use of existing opinion lexicons for determining sentiment polarity of WHY Questions asked on product review sites and evaluate their efficiency.
2. We address feature based sentiment analysis of questions.
3. We address opinion mining from complex comparative sentence. We propose a method that extracts main focused product of the author from comparative questions.

Rest of the paper is organized as follows. Section 2 deals with a review about 'WHY' QAS. Section 3 discuss about proposed approach for determining sentiment polarity of WHY Questions. We conclude and identify future scope in Section 4.

2 Related Work

Based on works on opinion question answering [1, 2, 4, 5, 6, 10, 14,15], we find that sentiment polarity analysis of an opinion question is the key component in drawing answers to opinion 'WHY' questions. Sentiment polarity of opinion questions is de-

termined through identification of opinionated words and computation of their polarity score through opinion lexical resources [1, 2, 4, 5, 6, 14, 15].

S. Moghaddam et al consider only adjectives as opinionated words for the task of determining sentiment polarity of questions [4,8]. Jong huet al. consult a Japanese polarity dictionary in their question answering [2].The dictionary is not available in English. Jianxing Yu et al. [5] develop an OPQA and determine sentiment polarity of questions with the use of MPQA sentiment lexicon [7]. Most of the words in MPQA project are objective words such as buy; purchase, choose etc. Hence, we consider the corpus as not a good choice.

OpinionFinder [7] performs document level analysis and identify subjective sentences and sentiment expressions in the text. The document level analysis is not appropriate for questions that have multiple opinions on different features of products.

The existing average scoring methods approach [1, 2, 4, 5, 6, 14, 15] could yield false results in determining sentiment polarity of questions e.g. WHY Nokia is good but Micromax is a bad mobile?

Ganapathibhotlaet al. [11] perform Opinion Mining in Comparative Sentences with an assumption that objects generally appear on both sides of a comparative word. This is not true in some cases like WHY it is better to have Nokia over Samsung?;"WHY there are more favorable comments about Nokia than Micromax"

In our previous work [15], we use semantic role labeler to identify main focused product from comparative sentences in WHY question answering. The method do not give promising results on some complex questions like "If I need a great mobile that could handle basic computing needs. WHY should I go for Samsung over Nokia?", "WHY Dells are ok, HPs aren't that good, but Macs are Fantastic", "WHY there are more favorable comments about Nokia than Micromax", "WHY people are incredibly more dissatisfied with battery backup of Nokia in comparison to Samsung " etc.

Stanford Sentiment [12] has not shown good performance on opinion WHY questions asked on product review sites [19].

From the literature surveyed in this section, we find that sentiment analysis of questioners from complex comparative WHYquestions arestill an issue.

3 Proposed Approach

In this section, we determine sentiment polarity of questioners. The recent work [1, 14, 15, 16] which does average scoring of words could yield false results in determining sentiment polarity of questions e.g. WHYNokia is good, Samsung is okay but Micromax is a bad mobile.

Our aim is to find sentiment of a questioner based on interpretation of the WHY question asked by him or her.

There is a need to split a WHY- question into different text segments based on different features of products. We further compute sentiment polarity of the question.

3.1 Splitting of WHY-Questions into Different Segments

The objective is to identify the key opinionated spansbased on individual feature or product within a WHY question. For example, in question, WHY Dells are ok, HPs are good, but Macs are Bad. Thetext segments are: (Dells are ok); (HPs are that good); (Macs are bad).

In this regard, we parse the sentence of question through a parser [18]. We examine the output of the parser and create a rule for finding the key opinionated spans. Rule is as follows:

Extract semantic arguments connected with each verb (v1, v2 and so on).

1. The arguments will be in the form of (arg1, verb_arg12, arg2) andname such groups as Group 1, 2 and so on.
2. If there is a word adj_arg1 thatis semantically related to arg1, weannotate the element (adj_arg1(arg1)).Same is done for verb_arg12, and arg2.
3. If the arg1 or arg2 is a phrase and there is verb in the phrase, we repeat the process of tagging from step 1.

Based on the analysis, we form each group as [{adj_arg1 (arg1) - adj_verb (verb) - adj_arg2 (arg2)}.{adj_arg1 (arg1) - adj_verb (verb) - adj_arg2 (arg2)}.{adj_arg1 (arg1) - adj_verb (verb) - adj_arg2 (arg2)}].

Separate groups (Group 1, 2 and so on) are created for all other verbs present in the question.

3.2 Computation of sentiment polarity of each group

We compute sentiment polarity of each group through following steps as discussed below:

- **Computing score of Opinion word**: we compute the score of each opinion word of each group.
- **Computing score of Group**: we finally use the scores of all words in the group to perform sentiment analysis of questioners.
- **Computing score of Opinion word:** We follow proposed approach used in [15] to compute sentiment polarity of each word. We make combined use of MPQA subjectivity Lexicon and SentiWordNet. Details are given in [15].
- **Sentiment analysis based on Features:** We manually compile list of products and their features collected from different review websites.

So the products and their features are already known.We compute the sentiment polarity of each group.

- Case 1: If in a group g1, there is only one feature then, the sentiment polarity of the group provide opinion on the feature (as positive and negative).
- Case 2: If there are more than one features f1 and f2 in a group. Firstly, we search for adj_arg1 and arg1 relationship where any feature could be arg1. Final score is computed in following manner:

(a) If both adj_arg1 and arg1 are positive, then the combination is made positive with score that is largest in magnitude of the two arguments.
(b) Else if both adj_arg1 and arg1 are negative, then the combination is made positive with score that is largest in magnitude of the two arguments.
(c) Else, the combination is made negative with score that is largest in magnitude of the two arguments.

In this way, we have two separate scores for f1 and f2 as SF1 and SF2.

Secondly, we compute score of the group based on F1 and F2.

We compute the scores (S) of other members i.e., members except (adj_arg1 f1, adj_arg1 f2).

Feature1 = aggregate score of (S,SF1)
Feature2 = aggregate score of (S,SF2)

In a question: WHYNokia is a great product for a good price when compared with Samsung?, we see that there are more than one feature (product, price) found in the group. We find semantic relation as:

- Nokiais (a [(great) product] for a [(good) price])
- score of product = score (Nokia + is +a+ great)
- score of price = score (Nokia +is+a+ for+good)

Some examples are given below in Table 1.

Table 1. Identification of feature related expression of product from questions

Examples	Group1	Group2	Group3
WHY Dells are ok, HPs aren't that good, but Macs are Fantastic	Dells are ok -Arg1 arg_12 arg2	HPs are (that good) Arg1 arg_12 arg2 Not are Adj_arg1 arg1	Macs are Fantastic -Arg1 arg_12 arg2

As per our algorithm,

- In question 1: We have group 1 as Dells areok ; group 2 as HPs (not are) (that good); group 3 as Macs are Fantastic.
- In question 2: We have group 1 as (The price and features) is (the [(main) concern]); group 2 as I neglect Nokia.
- In question 3: We have group 1 as I need (a mobile with [(ergonomic) design]); group 2 as I recommend Nokia
- In question 4: We have group 1 as Nokiais (a [(great) product] for a [(good) price])

3.2 Computing Sentiment Polarity of WHY Type Questions Using Existing Systems

We determine the polarity of forty Questions through the existing systems [Question Data Set is given after reference section under heading "Question Set for computing sentiment polarity"]. We compare six systems in Table 2.

Table 2. Performance of popular existing systems for sentiment analysis of questioners (on 40 questions given in Appendix 1).

System	MPQA [7]	SentiWordNet [21]	Wordnet [21]	Bing Liu [21]	SenticNet [16]	SentiStrength [18]
Accuracy	65.5%	65.5%	40%	60%	65.5%	67.5%

We perform comparative analysis of different lexicons used in our proposed method in Table 3.

Table 3. Performance of our method using different lexicons (on 40 questions given in Appendix 1)

Dictionary used in our method	SentiWordNet [21]	MPQA Lexicon [7]	Bing Liu Opinion Lexicon [21]	SenticNet 3.0 [18]
Accuracy	75%	57%	47%	67%

3.3 Addressing comparative opinions

We analyze 39 different forms of comparative questions (see Appendix 2). Following rules are followed to extract main focused product.

1. If in a group, there is only one product then, the only product is main focused product.
2. If in a group there are more than one products described in a group, then we follow rules as follows:

   – If there is a semantic relation: arg1 prep_arg12 arg2. Moreover, a product p1 is present in arg1 then, we select the product p1 as main focused product.
   – Else if there a semantic relation: arg1 prep_arg12 arg2 where a product p2 is arg2 and a word w1 is arg1. The word w1 has semantic relation: adj_arg12 with arg1 (product p1). Product p1 is selected as main product.

The performance of our method and existing methods on Question Set is presented in Table 4.

Table 4. Performance of existing systems in finding main focus

Method for finding main focus	Bing Liu	Mishra et al [15]	Ours
Accuracy	58.97%	71.7%	94.87%

4 Conclusions and Future Works

In this paper, we determine the sentiment analysisof the questionersfrom multi featured complex questions through proposed algorithm. We perform semantic analysis of WHY type questions and identify opinionated spans before computing sentiment polarity of question. The segmentation of WHY-questions isreliant on performance of automatic Enjuparser. Instead of computing score of each word of the group, we examine that find relevant opinion words and using their scores could enhance the accuracy of 'WHY' QAS. We comprehend thatSenticNet, SentiWordNet, MPQA are general opinion lexicons. There is a need for domain specific lexicons for effective opinion mining.

Our future work will be to employ ranking strategies for ranking features desired from question by questioners. We will exploit machine learning methods for the sentiment analysis of questioners. We will work on sentiment analysis of questioners posing questions in Arabic language.

References

1. Hongping Fu et al., "Classification of opinion questions", Proceedings of the 35th ECIR 2013 conference, pp. 714-717, Springer.
2. Jong-Hoon Oh et al. "WHY-question answering using sentiment analysis and word classes", In Proceedings of EMNLP-CoNLL 2012.
3. Chihli Hung, Hao-Kai Lin, "Using Objective Words in SentiWordNet to Improve Sentiment Classification for Word of Mouth," IEEE Intelligent Systems, 08 Jan. 2013.
4. S. Moghaddam and M. Ester, "AQA: Aspect-based Opinion Question Answering", IEEE-ICDMW, 2011.
5. Yu J, Zha Z-J, Wang M, Chua T-S, "Answering opinion questions on products by exploiting hierarchical organization of consumer reviews", in Proceedings of the conference on empirical methods on natural language processing (EMNLP), Jeju, Korea, pp 391–401, 2012.
6. L.W. Ku, Y.T. Liang, and H.H. Chen, "Question Analysis and Answer Passage Retrieval for Opinion Question Answering Systems", International Journal of Computational Linguistics & Chinese Language Processing, 2007.
7. T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-level Sentiment Analy-sis.HLT/EMNLP, 2005.
8. S. Moghaddam and F. Popowich."Opinion polarity identification through adjectives" CoRR, abs/1011.4623, 2010.
9. A PDTB-Styled End-to-End Discourse Parser developed by Ziheng Lin et al. http://wing.comp.nus.edu.sg/~linzihen/parser/
10. Fan Bu, "Function-based question classification for general QA", Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1119–1128, Massachusetts, USA, 9-11 October 2010.
11. Ganapathibhotla, Murthy, and Bing Liu. 2008. "Mining Opinions in Comparative Sentences". In Proc. of the 22nd International Conference on Computational Lin-guistics, Manchester.
12. Stanford sentiment analysis http://nlp.stanford.edu:8080/sentiment/rntnDemo.html.

13. Anders Björkelund, Love Hafdell, and Pierre Nugues. *Multilingual semantic role labeling*. In Proceedings of The Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pages 43--48, Boulder, June 4--5 2009.
14. Amit Mishra, Sanjay Kumar Jain, "An Approach for Computing Sentiment Polarity Analysis of Complex Why-type Questions on Product Review Sites", Research in Computing Science, 84, 2014, pp. 65-76.
15. Amit Mishra et al., "An Approach for Sentiment Analysis of Complex Comparative Opinion WHY Type Questions Asked on Product Review Sites", in Proceedings of CICling 2015, International Conference on Intelligence Text Processing and Computational Linguistics, Springer LNCS.
16. E. Cambria, A. Hussain. "Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis". Cham, Switzerland: Springer (2015)
17. E. Cambria, D. Olsher, and D. Rajagopal. SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. In: AAAI, pp. 1515-1521, Quebec City (2014).
18. R. Higashinaka and H. Isozaki. "Corpus-based Question Answering for 'WHY'-Questions". In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), pp. 198–425, 2008.
19. S. Verberne, L. Boves, N. Oostdijk, and P.A. Coppen. "What is not in the Bag of Words for 'WHY'-QA?" Computational Linguistics, 2010.
20. Enju Parser url :http://kmcs.nii.ac.jp/~yusuke/enju/demo.html.
21. Esuli and Proceedings of LREC Sebastiani, F., "SentiWordNet: A publicly available lexical resource for opinion mining", In -06, the 5th Conference on Language Resources and Evaluation, Genoa, Italy, 2006.

Appendix 1. Question Set for Computing Sentiment Polarity

3. WHY should I buy Nokia if I need mobile with good looks and nice sound quality?
4. WHY should I look for Nokia if I need mobile with good looks and nice sound quality?
5. WHY should one feel sad after buying X?
6. I need mobile with good sound quality and nice looks. WHY should one feel sad after buying x?
7. If I need mobile with good looks and nice sound quality, WHY should I insist Nokia?
8. WHY Nokia should be good option when we need a mobile?
9. WHY Nokia is good pick as a mobile?
10. WHY should one regret for long time after buying Nokia?
11. I went to market because I need mobile with good camera. WHY Should I go for Nokia?
12. WHY I bought Nokia at cheaper price but feel cheated?
13. WHY should one suggest Nokia as an alternative to x?
14. I went to market and bought Nokia. WHY should I feel satisfied finally?
15. WHY I went to market for buying Nokia?
16. I went to shop. I heard good things about Nokia. Hence I bought it. WHY Should I be happy?

17. If I need Nokia then WHY Nokia lumia is first choice?
18. WHY one feel cheated in the end after spending money on Nokia?
19. WHY one gets sick but need Nokia for daily purpose?
20. WHYshould one is inclined towards Nokia next after getting salary?
21. I went to shop. I took money from atm. I want good mobile. WHY should I order Nokia?
22. WHY should one buy Nokia instead of looking for its bad reviews?
23. If the price is another driving influence for purchasing mobile for me, I should choose Nokia over Samsung?
24. WHY Nokia is a great product when compared with Samsung
25. I would like at least a 4 hr. battery life. WHY should I advise Nokia over Samsung.
26. If I need a great mobile that could handle basic Computing needs. WHY should I go for Samsung over Nokia
27. WHY users cite negative reasons for those who prefer Nokia over Samsung
28. WHY people expressed positive opinions for Nokia as a better mobile when compared with Micromax
29. WHY one consider leaving x as alternative to y when there is requirement for good battery back up
30. WHY do Users cite negative feedback on mobiles manufactured by Nokia?
31. WHY people have objections against NAFTA
32. WHY should one aspire for Nokia instead of looking for its bad reviews?
33. WHY Nokia is wonderful, but very dangerous when it comes to emitting heat.
34. WHY people disapprove Nokia as best mobile in market.
35. WHY criticisms have been made about System of a Down or its music?
36. WHY people have the issues with Nokia Lumia
37. WHY Nokia is less competent in market when compared with Samsung
38. WHY people have objections toward Mayo or its products or research?
39. WHY Nokia is being considered worst than Samsung?
40. WHY people like Nokia better than Samsung?
41. WHY Nokia is good but Samsung is better mobile?
42. WHY it is better to neglect Nokia over Micromax?

Appendix 2. Questions Set for Finding Main Focus

43. WHY Dells are ok, HPs aren't that good, but Macs are Fantastic
44. The price and features is the main concern. WHY should I neglect Nokia over Samsung
45. I need a mobile with ergonomic design. WHY should I recommend Nokia over Samsung
46. My previous purchases were with Dell and HP. WHY should I prefer Nokia over dell and HP?
47. The price is another driving influence for purchasing mobile for me. I should propose Nokia over Samsung?
48. WHY Nokia is a great product for a great price when compared with Samsung

49. WHY people are incredibly more dissatisfied with battery backup of Nokia in comparsion to Samsung
50. I would like at least a 4 hr. battery life. WHY should I advise Nokia over Samsung.
51. If I need a great mobile that could handle basic computing needs. WHY should I go for Samsung over Nokia
52. WHY users cite negative reasons for those who prefer Nokia over Samsung
53. WHY people expressed positive opinions for Nokia as a better mobile when compared with Micromax
54. WHY there are more favorable comments about Nokia than Micromax
55. WHY Nokia is being considered worst than Samsung?
56. WHY people like Nokia better than Samsung?
57. WHYNokia is good but Samsung is better mobile
58. WHY fan following of bajrangi bhaijaan is cut by release of bahu bali
59. WHY one consider buying x as alternative to y when there is requirement for good battery back up
60. WHY should I love pictures taken by Nokia in place of Samsung
61. WHY should I give preference to Nokia over Micromax if I am looking for bad mobiles
62. WHY there are more favorable comments about Nokia than Micromax
63. WHY people say that it is better to have Micromax over Nokia in England than USA?
64. WHY people say that Nokia is better than Micromax in England than USA?
65. WHY people say that it is better to neglect Nokia over Micromax in England than USA?
66. WHY people say that Nokia is not as good as Micromax in England than USA?
67. WHY people say that Nokia is more valuable than Micromax in England than USA?
68. WHY people say that Micromax is good but Nokia is better in England than USA?
69. WHY people say that in market Nokia is more popular than Samsung in England than USA?
70. WHY people say that Nokia is much better than Samsung in England than USA?
71. WHY people say that Nokia is more efficient than Samsung to buy in England than USA?
72. WHY people say that people prefer Nokia over Micromax in England than USA?
73. WHY it is better to have Micromax over Nokia?
74. WHY Nokia is better than Micromax?
75. WHY it is better to neglect Nokia over Micromax?
76. WHY Nokia is not as good as Micromax?
77. WHY Nokia is more valuable than Micromax?
78. WHYMicromax is good but Nokia is better?
79. WHY in market Nokia is more popular than Samsung?
80. WHY Nokia is more efficient to buy than Samsung?
81. WHY people prefer Nokia over Micromax?

Investigation of the Feature Selection Problem for Sentiment Analysis in Arabic Language

Ahmed Nasser[1,2], Kıvanç Dinçer[1,*], Hayri Sever[1]

[1] Hacettepe University, Department of Computer Engineering, Çankaya Ankara, TR
[2] University of Technology, Control and System Engineering Department, Baghdad, IQ
{ahmed.r.nasser1984, kivanc.dincer, hayri.sever}@gmail.com

Abstract. Sentiment analysis, which is also known as opinion mining, can be defined as the process of the automatic detection of the attitude of an author towards a certain subject in textual contents. In this study we design and implement a document-level supervised sentiment analysis system for Arabic context and investigate its performance. We use three different feature extraction methods in order to generate three different datasets (unigrams, bigrams and trigrams) from the Opinion Corpus for Arabic (OCA). In order to find the optimal number of features and to obtain the best time performance in sentiment analysis, we employ two feature ranking methods (Information Gain based and Chi-Square based) and calculate the score of each feature with respect to the class labels. This feature ranking step selects only the features that are relevant to the class labels and removes the irrelevant features that cause unnecessary processing. Hence, it helps to increase the classification performance and reduce the processing time. Finally, we evaluate the performance of three standard classifiers for polarity on the previously generated unigram and bigram based data sets, namely Support Vector Machines, K-Nearest Neighbor and Decision Tree, known by their effectiveness over these types of datasets. In our study SVM classifier has showed superior classification performance compared to the other two classifiers. Our experimentation results also prove the effectiveness of the two feature selection methods we use in order to reduce the feature space of the generated datasets and provide higher classification performance.

Keywords: Arabic Sentiment Analysis, Machine Learning, Dataset Generation, Feature Selection Algorithms

1 Introduction

Sentiment analysis, which is also known as opinion mining, can be defined as the process of the automatic detection of the attitude (positive, negative or neutral) of an author towards a certain subject in textual contents by the use of natural language processing, text analysis and computational linguistics [1]. Sentiment analysis is currently considered among the most rapidly emerging research fields due to the immediate need of processing the opinionated web contents coming from social networks and web blogs.

There are many approaches in the literature that deals with the sentiment analysis of Arabic language [2]. The supervised or corpus-based approach employs different machine learning classifiers such as Support Vector Machine (SVM) [3], K-Nearest Neighbor (K-NN) [4], Decision Tree (D-Tree) [5] on some datasets to determine the sentiment orientation of the text [6]. In the alternative unsupervised approach or lexicon-based approach, special dictionaries are used to specify the polarity of a word or sentence [6]. There are also hybrid solutions combining the supervised and unsupervised approaches and they are called as weakly- or semi-supervised approaches [7].

It may be worth stating that especially the Subjectivity and Sentiment Analysis (SSA) subject has been receiving more attention among scholars [8, 9]. The SSA research includes: class prediction (i.e. subjective or objective); polarity prediction (i.e. positive, negative, or neutral) and even some level of classification (word, sentence or document level) [8].

We can summarize the goals of our study as follows: First, generating different datasets that can be used to support supervised sentiment analysis systems in Arabic context. Second, applying feature selection (reduction) techniques to reduce the feature space of the generated datasets which in turn leads the polarity classification performance to increase. Third, observing the performance of the implemented sentiment analysis system using different feature selection techniques and classifiers on the generated datasets.

We focused on implementing a supervised sentiment analysis system at document-level where a whole document should be classified as having either positive, negative or neutral polarity. We used the OCA Opinion Corpus for Arabic [1] as the base corpus for generating unigram, bigram and trigram-based data sets by applying different feature extraction techniques. We then applied different feature selection techniques [10] on those datasets to reduce the feature spaces. The features were selected using the statistical approaches such as Information Gain [11] and Chi-Square measure [11]. For the classification, we used three standard classifiers which are SVM, K-NN and D-Tree. Finally, the performance of the implemented system was experimented on the generated datasets using different testing scenarios to obtain the polarity classification performance metrics such as accuracy, precision, recall and f-measure with respect to the number of features.

In the next section, previous studies on the subject will be presented and discussed. In Section 3 we will explain the details of generating datasets as well as the method we proposed for evaluating those datasets. In Section 4 different test scenarios that are executed to evaluate the classification performance of the proposed system will be presented. In the last section we will finish the paper with the conclusion and presentation of the planned future work.

2 Related Work

Rushdi-Saleh et al. [1] proposed a document-level supervised sentiment analysis approach. They generated the Opinion Corpus for Arabic (OCA) from some movie and film reviews available on the web and used two classifiers (SVM and NB (Naïve Bayes)

[12]) to identify the polarity of those reviews. They generated three different n-gram schemes as features (unigrams, bigrams, and trigrams) and used TF–IDF (Term Frequency–Inverse Document Frequency) and TF (Term Frequency) weighting schemes in the validation process of a SVM-based sentiment classifier.

Shoukry and Rafea [13] used sentence-level supervised sentiment analysis approach for Arabic language by collecting the required data from Twitter. In the preprocessing phase, unigram and bigram based feature extraction techniques were applied and they concluded that the bigram model did not enhance the classification performance. They followed a corpus-based approach where SVM and NB with term frequency features were used for polarity classification. They extracted all the unigrams and bigrams in the corpus that exceeded a certain threshold.

Mountassir et al. [14] presented three methods to solve the imbalance issue in the dataset during SSA process: eliminate farthest, eliminate similar and eliminate by clustering. They used a document-level supervised SA approach. Two types of imbalanced corpus were generated from two different datasets: The Arabic dataset was collected from Al-Jazeera's website and the English dataset was collected from SINAI. They used a bags-of-words features representation with binary weighting. The proposed under-sampling techniques were applied on commonly used classification techniques such as NB, SVM, and K-NN and the results were comparatively analyzed.

Ahmed et al. [15] presented some of the challenges and issues that faced the SSA researchers in general and especially while working with the Arabic language, and proposed some solutions for them. They used a sentence-level supervised SA on the data collected from Twitter. They used different classifiers such as SVM, NB, Bayes Net [16] and J48 [17] to determine the polarity of Arabic tweets collected from different domains. They also tested the effects of different preprocessing techniques, feature extraction, and stemming methods on polarity classification. They used NB classifier with words N-grams frequency vector for sentiment classification.

Abdulla et al. [18] studied the two main approaches of sentiment analysis, namely sentence-level supervised and unsupervised SA for Arabic corpus collected from Twitter. They conducted different experiments using four well-known classifiers: SVM, NB, D-Tree, and K-NN. They used a sentiment lexicon to find the sentiment orientation of Arabic words and used the unigram technique for feature extraction. To determine the sentiment polarity of the entire input text, they aggregated the total sentiment scores of each individual word in that text.

Abdulla et al. [19], in another work of theirs, proposed a sentence level supervised SA with a large dataset consisting of Arabic comments. This dataset was manually collected and annotated from the "Yahoo! Maktoo" social network. They presented inclusive analysis of this dataset using NB and SVM classifiers with TF-IDF term weighting technique. Some additional information such as the number of likes or dislikes as well as the gender of the author were used to enhance the classification performance.

In [20] a sentence level supervised SA approach was presented. The dataset was collected from Arabic news websites such as Al Jazeera, BBC Arabic, Al-Youm Al-Sabe'a and Al Arabiya, Constitution Facebook Page, and People's Opinion Facebook page. They divided a "Slang Sentimental Words and Idioms Lexicon" (SSWIL) of opinion words into two classes: satisfaction and dissatisfaction classes. They proposed

a Gaussian kernel SVM classifier for Arabic slang language and used it to classify Arabic comments on Facebook.

El-Makky et al. [21] built a new Arabic lexicon by merging two Modern Standard Arabic MSA lexica, namely, MPQA [22] and ArabSenti [23] with two Egyptian Arabic lexica built from Twitter. They used both the sentence-level supervised and unsupervised SA approach. An augmented lexicon-based approach was used to define the semantic orientations (SO) of the words. The lexicon-based approach depends on the presence of opinion (or sentiment) words (looked-up from a sentiment lexicon). These words expressed positive or negative sentiments. The sentiment of the tweet that results from the modified algorithm was used as a semantic orientation score which was a component of the proposed feature vector. Subjectivity and polarity classifiers were used to classify the tweets.

3 Proposed Sentiment Analysis Approach

Our proposed Arabic supervised sentiment analysis approach consists of two parts. In the first part we generated different datasets using different feature extraction methods such as unigrams, bigrams and trigrams. These datasets were built based on the OCA Opinion Corpus for Arabic [1]. The second part of the system is the dataset evaluation and the supervised sentiment analysis system. In this part two feature selection methods were used to find the optimal number of features in each dataset in order to obtain a higher classification performance. For sentiment classification we used three different standard classifiers (SVM, K-NN and D-Tree). Each part of the proposed system is discussed in details bellow.

3.1 Dataset Generation

We use the OCA corpus for Arabic for generating the datasets that will be used later for building the classification model in our sentiment analysis system. The OCA corpus contains 500 text files where each file represents a movie review in Arabic language. The corpus data has been collected from 15 distinct web sites. Those 500 files are divided into two categories: positive and negative where each category consisted of 250 documents. Table 1 shows some statistics related to the OCA corpus.

Table 1. Statistics for the OCA Opinion Corpus for Arabic.

	Positive	Negative
Total word count in the corpus	130,981	104,080
Average word count for each file	524	416
Total unique word count in the corpus	79,262	66,066
Average unique word count for each file	317	265
Average sentence count for each file	12	18
Max sentence count	70	278
Max word count in a sentence	668	450
Average word count in a sentence for each file	154	75

Dataset generation process shown in Fig.1, consists of three main steps as follows:

- The first step is the preprocessing stage where each document in the corpus is first tokenized into their words and the stop words are removed. Then the root of each word is found using the Buckwalter morphological analyzer's Aramorph Arabic lemmatizer [24]. Finally, a sentiment filter is applied to these terms in order to remove the terms that are not presented in ArSenL [25]. ArSenL is a large scale standard Arabic sentiment and opinion-mining lexicon built using a combination of English SentiWordnet [26] and Arabic WordNet [27].
- The second step enumerates the terms from each document in the form of unigrams, bigrams and trigrams which represents the features that are used for generation of three separate datasets later.
- The last step does feature weighting. First, *TF-IDF* [28] values are calculated for each feature (term) in each document and used to construct the vector space model. The vector space model can be represented by an *(M × N)* matrix, where *M* is total number of documents and *N* is the total number of features in the corpus. This matrix contains a *TF-IDF* weight for each feature. The TF-IDF weights for each one of the previously generated unigram, bigrams and trigrams features will result in a separate data set, a total of three different datasets. Table 2 shows the total number of features for these three datasets.

Table 2. Total number of features in each dataset

Dataset	Method used	Total number of features
1	Unigram	26,270
2	Bigrams	134,871
3	Trigrams	267,451

3.2 Dataset Evaluation

The dataset evaluation process for the datasets generated in Section 3.1 is shown in Fig.2. Since the generated datasets contain large feature spaces, a feature selection method needs to be employed to reduce the number of features in each dataset. Either the Chi-square or the Information Gain (IG) method is applied to rank each feature according to its corresponding class information. Information gain is used as a metric to measure the reduction in entropy for the class category *(c)* prediction by knowing the presence or absence of a feature *(t)* in a dataset [11]:

$$IG(f, c_i) = \sum_{c \in \{c_i, \bar{c_i}\}} \sum_{\grave{t} \in \{t, \bar{t}\}} P(\grave{t}, c).log \frac{P(\grave{t}, c)}{P(\grave{t}).P(c)} \quad (1)$$

Chi-square [11] is employed to measure the lack of independence between a feature *(t)* and a class category (c_i) then compared to the chi-square distribution with one degree of freedom. Chi-square is defined as follows:

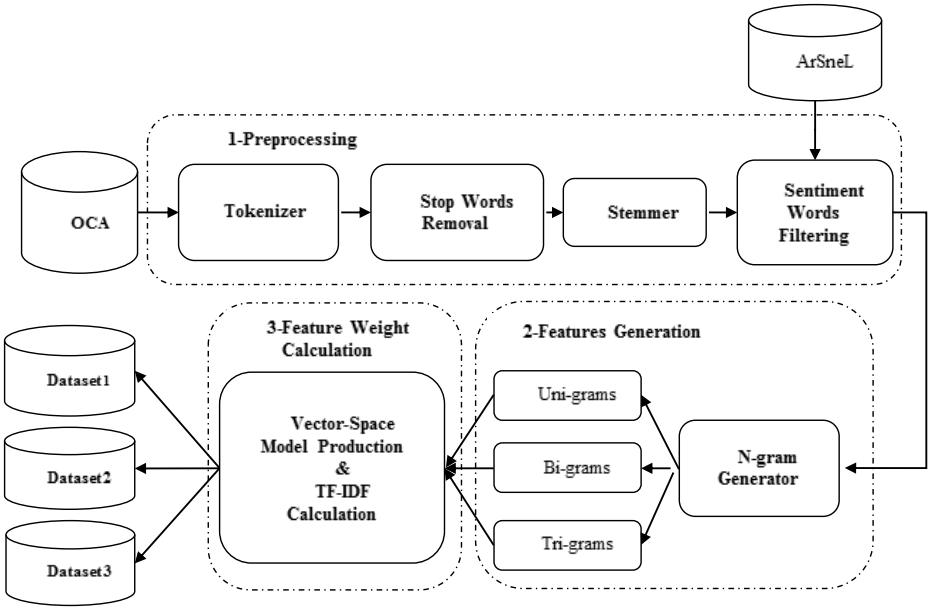


Fig. 1. Block diagram of our datasets generation method

$$X^2(t, c_i) = \frac{N[P(t,c_i)P(\dot{t},\dot{c}_i) - P(t,\dot{c}_i)P(\dot{t},c_i)]^2}{P(t)P(\dot{t},)P(\dot{c}_i)P(c_i)} \quad (2)$$

The features are ordered separately in descending order with respect to their information gain and Chi-square values.

In the evaluation process, the dataset is divided into two sub-datasets called training and testing sets. *K*-fold cross-validation [29] method is followed for randomly splitting the dataset into *K* (=10 in our case) equal subsets, and each time one of the *K* subsets is used as the testing set and the other *K-1* subsets are used as the training set.

The generated training set is used as input to three standard classifiers, SVM with linear kernel, K-NN with a cosine-based distance, and D-Tree. After the training process is completed, the testing dataset is applied over the trained classifiers and by comparing the results from classifiers with the class labels of the testing dataset the confusion matrix for each classifier is computed. From the confusion matrix, the classifier evaluation metrics such as accuracy, precision, recall and F-measure are calculated [29].

Before the evaluation process begins, the first 500 features (from the dataset with sorted features) are selected as input dataset to the evaluation process mentioned above and when it is done this means one iteration is complete. The number of selected features is increased by 1,000 in each iteration until the total number of features is reached.

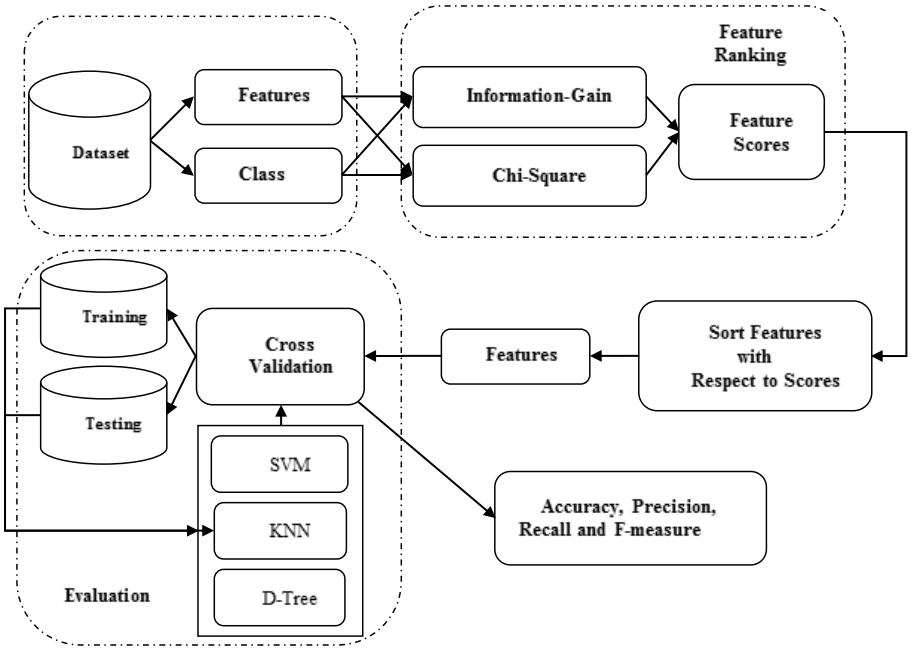


Fig. 2. The block diagram of our dataset evaluation system with the proposed feature selection approach.

4 Experimental Evaluation

At this stage we only evaluate two of the generated datasets, those that are based on unigrams and bigrams and will leave the evaluation of trigram-based dataset for the future work. The evaluation results are shown below for different scenarios:

4.1 First Testing Scenario: Testing with Unigrams

The dataset was generated from the OCA corpus using the vector space model, TF-IDF was used for term weighting, and the terms were all considered as unigrams. The total number of unigrams in the corpus are 26,270.

As stated previously, information-gain and chi-square coefficients are used to rank the features in descending order. Fig.3 shows the features sorted with respect to the information gain score.

The features are sorted in descending order with respect to their information gains and the chi-square score. The evaluation process starts with selecting the first 500 features and increasing it by 1,000 at each iteration until it reaches the total number of features. At each iteration *k*-fold cross validation method is used for generating the

training and testing sets, and each classifier is evaluated in 10 folds. For these 10 folds the average number of accuracy, precision, recall and F-measure metrics are calculated for each classifier. The results are shown in Fig.4 below.

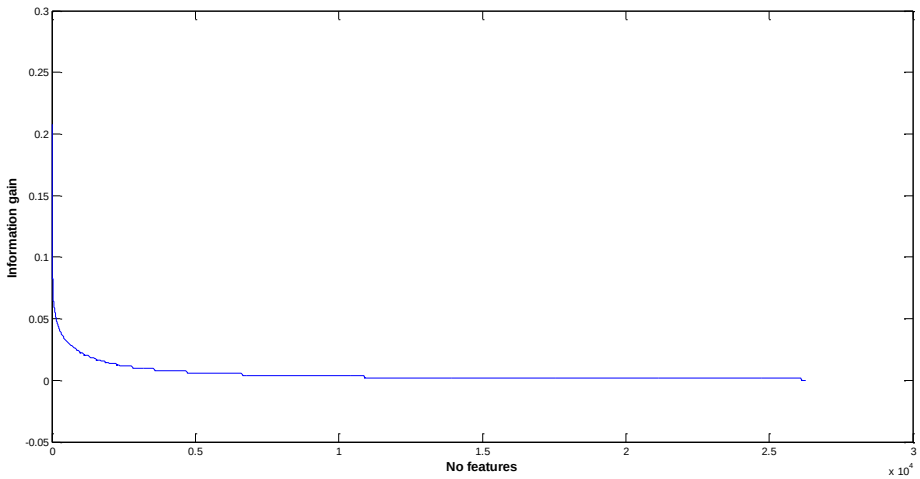


Fig. 3. Shows the features sorted with respect to its information gain score

As seen from the results presented in Fig.4, SVM classifier yields the best classification performance compared to K-NN and decision tree classifiers. D-tree classifier shows an average performance. K-NN shows the worst classification performance.

SVM classifier shows a very good performance in the region where the number of features are between 9,000 and 11,000. So the optimal number of features can be chosen as 10,000 out of the total 26,270 features and this gives 62% decrease in the feature space. Table 3 shows the performance metrics of SVM, K-NN and D-Tree classifiers with the selected 10,000 features using both information gain and chi-square score ranking.

Table 3. The performance metrics of the three classifiers using the unigram based dataset

	Information gain ranking			Chi-square score ranking		
	SVM	K-NN	D-Tree	SVM	K-NN	D-Tree
Accuracy	85	55	75	83	55	74
Precision	84	65	74	82	62	72
Recall	89	20	75	88	22	71
F-measure	88	30	75	88	34	74

4.2 Second Testing Scenario: Testing with Bigrams

The second scenario is similar to the first one with the exception that the generated dataset contains bigram based terms. The total number of bigrams for all documents in the corpus is 134,871.

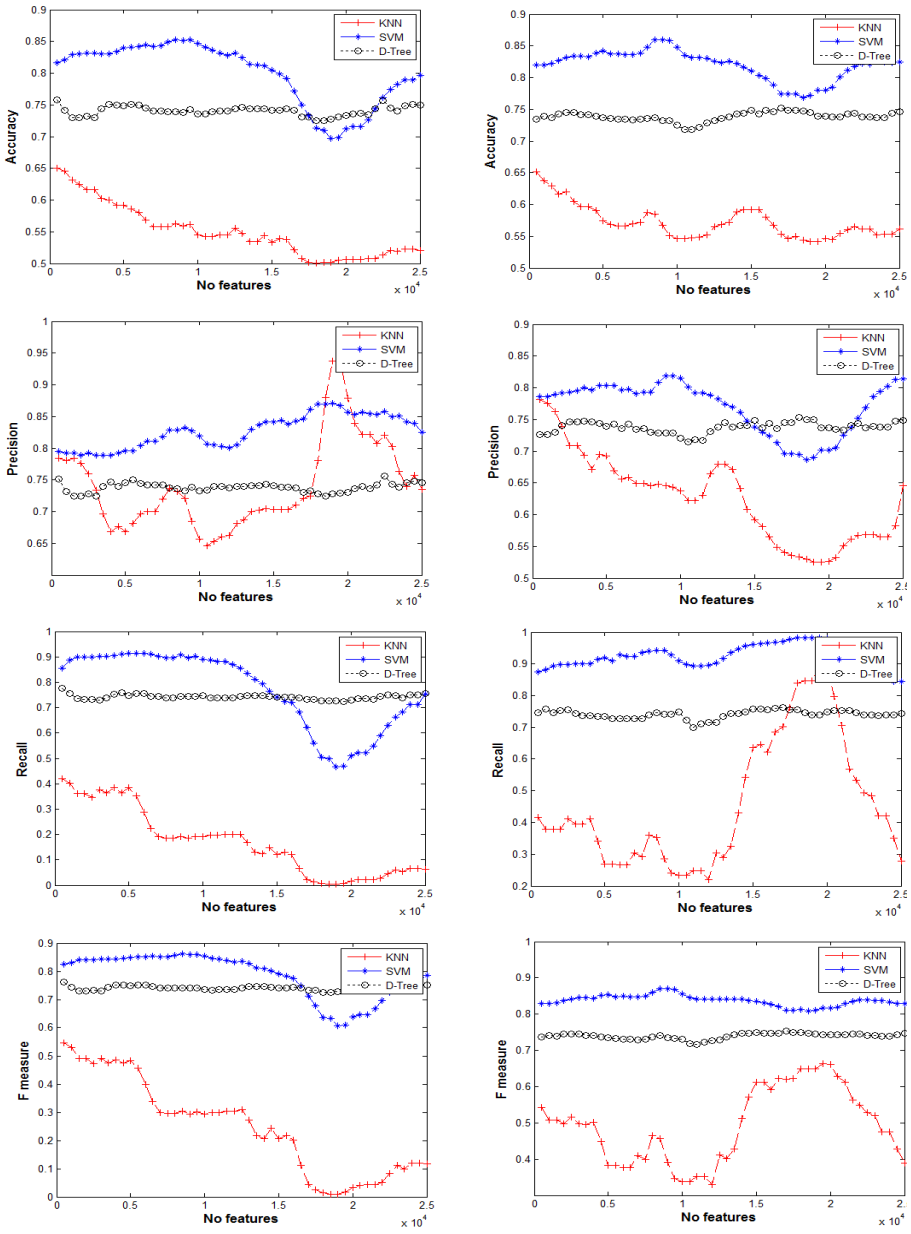


Fig. 4. (Accuracy, Precision, Recall and F-measure) versus the number of features for each classifier. The graphics on the left show the cases where the features were sorted with respect to the information gain, and the others show the features sorted with respect to the chi-square

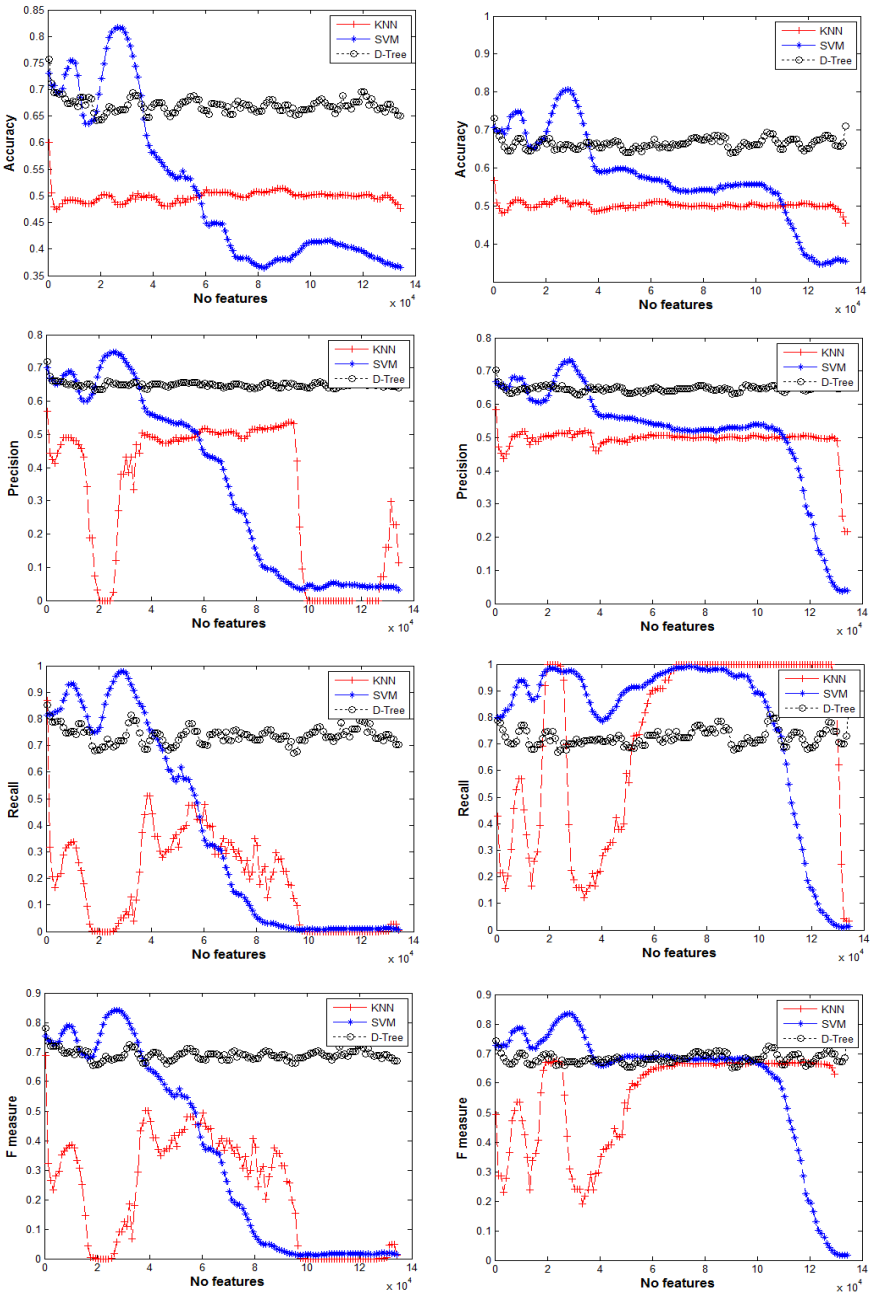


Fig. 5. (Accuracy, Precision, Recall and F-measure) versus the number of features for each classifier. The graphics on the left show the cases where the features were sorted with respect to the information gain, and the others show the features sorted with respect to the chi-square

After applying the same evaluation process described in the first scenario on the bigram based dataset, we get the classifier evaluation results shown in Fig.5.

SVM classifier shows the best classification performance among the tree, but as the number of features increases its performance drops. The D-tree classifier's performance seems to be stable with the increasing number of features. K-NN shows the worst classification performance.

SVM classifier shows the maximum performance in the region where the number of features is between 20,000 to 40,000. Therefore, the optimal number of features can be chosen as 30,000 out of the total of 134,871 features and this gives 78% decrease in the feature space. Table 4 shows the performance metrics for each classifier (SVM, K-NN and D-Tree) with the selected 30,000 features using both information gain and chi-square score ranking.

Table 4. The performance metrics of the three classifiers using the bigram based dataset

	Information gain ranking			Chi-square score ranking		
	SVM	K-NN	D-Tree	SVM	K-NN	D-Tree
Accuracy	80	49	63	75	53	64
Precision	78	50	65	70	50	61
Recall	97	15	79	96	20	65
F-measure	81	16	65	79	19	66

5 Conclusion and Future Work

In this study, we focused on implementing a document-level supervised sentiment analysis systems in Arabic context. We first generated three different datasets based on OCA Opinion Corpus for Arabic. These datasets were generated using different feature extraction methods (unigrams, bigrams and trigrams) with TF-IDF feature weighting. Then we used a supervised sentiment classification system that emphasize the Information Gain and Chi-square methods for feature selection. The value of the Information Gain and Chi-square coefficient was used for ranking of each feature in the datasets. The feature selection method aims to find the optimal number of features in each dataset that provides an optimum (higher) classification performance. Then, three of the standard classifiers (SVM, K-NN and D-Tree) were used for the sentiment classification.

We designed several testing scenarios for evaluating the performance of our proposed sentiment classification evaluation system. At each testing scenario all three classifiers and both feature selection methods were applied on each one of the previously generated datasets. We collected the (Accuracy, Precision, Recall and F-measure) classification performance metrics for each classifier using the k-fold cross validation. That is, we experimented with the generated unigram and bigram based datasets and measured the performance of each classifier (SVM, K-NN and D-Tree) using both Information Gain and Chi-square based feature selection methods.

The results show that SVM based sentiment classifier provides the best classification performance among three. The K-NN is considered as the worst classification performance, while the D-Tree classifier shows an average performance.

In our case, using the feature selection methods, the unigrams based dataset with SVM classifier shows very good performance (up to 88% in F-measure) where the number of features are approximately 10,000 out of 26,270 features. Thus this is a 62% decrease in the feature space. The bigrams based dataset shows the maximum performance (up to 81% in F-measure) using SVM classifier when the number of features is approximately 30,000 out of 134,871 total features hence, this gives 78% decrease in the feature space, providing much higher classification performance.

The advantages of SVM classifiers over the other classification methods have been reported by other researchers too. "SVM is robust in high dimensional feature spaces, works very good if any feature is relevant, data is linearly separable and most text categorization problems are considered as linearly separable" [1, 30]. It is very remarkable that SVM is superior to many other machine learning techniques [1, 13, 15, 18, 19, 20, 21].

In the future, we plan the following actions: a) Evaluating the third dataset which uses trigrams as features using the proposed evaluation system, and use Naïve-bias and neural network [31] based classifiers to evaluate the generated datasets. b) Trying other feature selection methods (such as Correlation coefficient, Odds ratio [11] and minimum-redundancy maximum-relevancy "mRMR" [32]) and showing the impact on the results of the generated datasets. c) Using hybrid features which are generated by combining more than one feature (i.e. combining TF-IDF term weighting features with term sentiment score features that can be obtained from ArSenL sentiment lexicon).

References

1. Rushdi-Saleh, M., Martín-Valdivia, M.T., Ureña-López, L.A., Perea-Ortega, J.M.: OCA: Opinion corpus for Arabic. Journal of the American Society for Information Science and Technology. 62, 10, 2045–2054 (2011)
2. Farra, N., Challita, E., Assi, R.A., Hajj, H.: Sentence-Level and Document-Level Sentiment Mining for Arabic Texts. In: 2010 IEEE International Conference on Data Mining Workshops. pp. 1114–1119 IEEE (2010)
3. Fung, G.M., Mangasarian, O.L.: Multicategory Proximal Support Vector Machine Classifiers. Machine Learning. 59, 1–2, 77–97 (2005)
4. Dasarathy, B.: Nearest Neighbor (NN) Norms : NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (Cal.) [etc.] (1991).
5. White, A.P., Liu, W.Z.: Technical Note: Bias in Information-Based Measures in Decision Tree Induction. Machine Learning. 15, 3, 321–329 (1994)
6. Read, J., Carroll, J.: Weakly supervised techniques for domain-independent sentiment classification. In: Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion - TSA '09. p. 45 ACM Press, New York, New York, USA (2009)
7. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics. 37, 2, 267–307 (2011)
8. Montoyo, A., Martínez-Barco, P., Balahur, A.: Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments. Decision Support Systems. 53, 675–679 (2012)
9. Abdul-Mageed, M., Diab, M., Kübler, S.: SAMAR: Subjectivity and sentiment analysis for Arabic social media. Computer Speech & Language. 28, 1, 20–37 (2014)

10. Sarakar, S., Goswami, S.: Empirical Study on Filter based Feature Selection Methods for Text Classification. International Journal of Computer Applications. 81, 6, 38–43 (2013)
11. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. ACM SIGKDD Explorations Newsletter. 6, 1, 80 (2004)
12. Pazzani, M., Domingos, P.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning. 29 (2–3). p.p 103–130 (1997)
13. Shoukry, A., Rafea, A.: Sentence-level Arabic sentiment analysis. In: 2012 International Conference on Collaboration Technologies and Systems (CTS). pp. 546–550 IEEE (2012)
14. Mountassir, A., Benbrahim, H., Berrada, I.: An empirical study to address the problem of Unbalanced Data Sets in sentiment classification. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 3298–3303 IEEE (2012)
15. Ahmed, S., Pasquier, M., Qadah, G.: Key issues in conducting sentiment analysis on Arabic social media text. In: 2013 9th International Conference on Innovations in Information Technology (IIT). pp. 72–77 IEEE (2013)
16. Friedman, N., Geiger, D., & Goldszmidt, M.: Bayesian network classifiers. Machine learning, 29(2-3), p.p 131-163 (1997)
17. Clark, P., Boswell, R.: Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, San Francisco (2000)
18. Abdulla, N.A., Ahmed, N.A., Shehab, M.A., Al-Ayyoub, M.: Arabic sentiment analysis: Lexicon-based and corpus-based. In: 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). pp. 1–6 IEEE (2013)
19. Abdulla, N.A., Ayyoub, M. Al, Kabi, M.N. Al: An extended analytical study of Arabic sentiments. International Journal of Big Data Intelligence. 1, 1/2, 103 (2014)
20. Soliman, T., Soliman, T.H., Elmasry, M.A., Hedar, A., Doss, M.M.: Sentiment Analysis of Arabic Slang Comments on Facebook. International Journal Of Computers & Technology. 12, 5, 3470–3478 (2013)
21. El-Makky, N., Nagi, K., El-Ebshihy, A., Apady, E., Hafez, O., Mostafa, S., Ibrahim, S.: Sentiment Analysis of Colloquial Arabic Tweets. In: ASE BigData/SocialInformatics/PASSAT/BioMedCom 2014 Conference, Harvard University. pp.1–9 (2014)
22. Mihalcea, R., Banea, C., Wiebe, J.: Learning multilingual subjective language via cross-lingual projections. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. 976–983 (2007)
23. Abdul-Mageed, M., Diab, M.: SANA: A Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis. Proceedings of the Language Resources and Evaluation Conference. 1162–1169 (2014)
24. Buckwalter T. Buckwalter Arabic Morphological Analyzer version 1.0, Philadelphia: Linguistic Data Consortium, Catalog No.LDC2002L49, ISBN 1-58563625760, 2002.
25. Badaro, G., Baly, R., Hajj, H., Habash, N., El-Hajj, W.: A large scale Arabic sentiment lexicon for Arabic opinion mining. In: ANLP 2014, p. 165 (2014)
26. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3 . 0 : An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining SentiWordNet. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation. pp. 1–12 (2010).
27. Black, W., Elkateb, S., Pease, A., Rodriguez, H., Alkhalifa, M.: Introducing the Arabic WordNet Project. Word Journal Of The International Linguistic Association. 22, 295–299 (2006)
28. Sebastiani, F., Fabrizio: Machine learning in automated text categorization. ACM Computing Surveys. 34, 1, 1–47 (2002)
29. Mitchell, T: Machine learning. MIT Press. (1997)

30. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 137–142 Springer Berlin Heidelberg (1998)
31. Dhande, L. L., Patnaik, P. G. K.: Analyzing Sentiment of Movie Review Data using Naive Bayes Neural Classifier. 3(4), 313–320 (2014)
32. Peng, H. C., Long, F. H., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence. 27, 8, 1226–1238 (2005)

A Proposed Lexicon-Based Sentiment Analysis Approach for the Vernacular Algerian Arabic

M'hamed Mataoui,[1] Omar Zelmati,[2] Madiha Boumechache[2]

[1] IS&DB laboratory, Ecole Militaire Polytechnique, Algiers, Algeria
[2] Ecole Militaire Polytechnique, Algiers, Algeria
mataoui.mhamed@gmail.com, omar.zelmati@gmail.com,
madiha.boumechache@gmail.com

Abstract. Nowadays, sentiment analysis research is widely applied in a variety of applications such as marketing and politics. Several studies on the Arabic sentiment analysis have been carried out in recent years. These studies mainly focus on Modern Standard Arabic among which few studies have investigated the case of Arab dialects, in this case, Egyptian, Jordanian, and Khaliji. In this paper, we propose a new lexicon-based sentiment analysis approach to address the specific aspects of the vernacular Algerian Arabic fully utilized in social networks. A manually annotated dataset and three Algerian Arabic lexicons have been created to explore the different phases of our approach.

Keywords: Arabic sentiment analysis, vernacular Algerian Arabic, Algerian dialect, Modern Standard Arabic, Social networks.

1 Introduction

The last years are mainly characterized by the fast proliferation of social networking services such as Facebook, Twitter and YouTube. These social networks allowed individuals and groups to express and share their opinions about different kinds of topics (products, political events, economics, restaurants, books, hotels, video clips, etc.). Billions of comments and reviews are added to the web each day, which has led to the need to mine users' opinions in order to discover useful information. Mining this enormous volume of comments and reviews is almost impossible manually. Therefore, a new thematic of Natural Language Processing (NLP), known as sentiment analysis (SA) or opinion mining (OM), emerged. The main purpose of sentiment analysis is to extract users' sentiments/opinions from created contents by using automatic mining techniques to determine their attitudes with respect to some topic, often expressed in textual form.

Nowadays, sentiment analysis is used mainly by businesses to discover the opinions of different customers as part of marketing purposes [1, 2]. It is also used in politics to predict election results or to know public opinions about different policies. SA field is considered as a classification task for deciding about an opinion as being positive, negative, or neutral.

According to [3], SA approaches are based on one of the two following classes: lexicon-based approaches [4, 5]; corpus-based approaches [6, 7].

Most of existing research on sentiment analysis focuses on English text [2, 4, 8]. In spite of its importance as one of the most used languages in the world, only a limited number of research on Arabic sentiment analysis has been carried out. The proposed Arabic sentiment analysis approaches mainly focus on Modern Standard Arabic (MSA) [1, 9], among which few studies have investigated the case of Arab dialects (colloquial Arabic), namely, Egyptian [1, 10-12], Jordanian [1, 9], and Khaliji (dialect used in the Gulf countries) [13]. To our knowledge, research on sentiment analysis for the Maghreb dialects or Maghrebi Arabic (Algerian, Moroccan and Tunisian) is almost non-existent [14].

The purpose of this work is to begin a reflection to study the sentiment analysis for the case of the Algerian dialect, very different compared to other Arabic dialects, not only in pronunciation, but rather by its different textual forms, very diverse and extremely rich.

This paper is organized as follows: In Section 2, related work is presented. Section 3 presents the peculiarities of vernacular Algerian Arabic. Section 4 describes our sentiment analysis approach and presents our different datasets and experimental results obtained. Finally, in Section 5, we conclude with some prospects.

2 Related Works

In this section, we will present research related to Arabic sentiment Analysis field with focus on dialectal Arabic study cases.

Arabic language is characterized by a wide number of dialects varieties. Besides Modern Standard Arabic used as a formal language, different Arabic dialects are used for nearly all everyday speaking situations. By the emergence of social media and the various electronic networks, enabling Arab users to express their opinions using different Arabic dialects, researchers have raised the need to consider this amount of generated content especially by the study of the peculiarities related to written forms of these different dialects.

Research on Arabic sentiment analysis field can be classified into three categories: First, the work that are interested in building SA related resources (corpus and lexicons). Secondly, the work which propose sentiment analysis approaches for MSA and Arabic dialects (lexicon-based, corpus-based). Finally, work which propose techniques related to SA improvement (pre-processing, morphological analyzers, etc).

2.1 Building resources for Arabic sentiment analysis

Abdul-Mageed and Diab [15] constructed a large-scale multi-lingual lexicon based on both MSA and colloquial Arabic (Egyptian and Levantine) for sentiment analysis, called SANA. SANA lexicon is a combination of many lexicons, such as, SIFAAT, HUDA and an automatic collected corpora (with both statistical method and machine translation).

Abdulla Nawaf et al. [13] addressed the issue related to the lack of freely accessible datasets for analysis and testing in the Arabic sentiment analysis context. A relatively large dataset of Arabic comments and reviews from Yahoo!-Maktoob social network has been manually collected and annotated. The annotation phase was made by two or three experts in MSA and Jordanian dialect. Authors have used two classifiers (SVM and Naive Bayes) in their experiments. They have showed that SVM outperforms NB and achieves about 64% of accuracy level [13].

Gilbert Badaro et al. [16] addressed issues related to the build of an Arabic sentiment lexicons. They produce ArSenL, the first publicly available large scale Arabic sentiment lexicon. ArSenL is based on a combination of existing resources, like: ESWN, Arabic WordNet, and SAMA. Their experimental study shows that using English-based linking produces superior performance in comparison to using the WordNet-based approach. Authors showed also that the combination of the two resources is better than either.

Diab et al. [17] developed an electronic lexicon that can be used in different NLP tasks, sentiment analysis in our case. Their lexicon consists of three parts: MSA, dialectal Arabic and English. Authors made Tharwa publicly available which can be used mainly for the Egyptian dialect sentiment analysis.

Al-Kabi et al. [18] shows the creation of a flexible and relatively big corpus, that consists of 250 topics equally divided among five domains (economy, food-life style, religion, sport, and technology), for Arabic sentiment analysis. Their manually created corpus is characterized by its flexibility and is constituted mainly of comments and reviews expressed in both MSA and Colloquial Arabic. It contains five types of reviews (English, mixed MSA & English, French, mixed MSA & Emoticons, and mixed Egyptian & Emoticons). Authors show in their analysis that most of the users of Yahoo! Maktoob prefer to use of MSA. In addition, the created corpus contains few comments and reviews that used English, French, Emoticons, etc.

2.2 Sentiment analysis approaches for MSA and Arabic dialects

Itani et al. [19] conducted a comparison between the lexicon-based and corpus-based approach by using both MSA and Arabic dialects. The experimental results shows that lexicon-based approach (83.4% of accuracy) outperforms the corpus-based approach.

Ahmad et al. [20] are the first who investigated Arabic sentiments analysis by studying the case of financial news. They showed that the proposed local grammar approach, developed on an archive of English texts can be applied to both Chinese and Arabic languages.

El-Beltagy et al. [21] focused on problems, challenges and open research issues related to Arabic sentiment analysis. They proposed to build domain-based and vernacular-based Arabic sentiment lexicons and consider the computation of the semantic orientation of Arabic Egyptian tweets as a case study for which an Egyptian dialect sentiment lexicon has been created. They used two methods of polarity computation: straightforward sum and double polarity sum. Experimental results showed that the use of weighted lexicons with double polarity sum obtained good improvements.

Al-Kabi et al. [1] proposed to build a sentiment analysis tool for colloquial Arabic and MSA, called: CNSA-MSA-SAT. They collected a large number of comments and reviews to build polarity lexicons used by CNSA-MSA-SAT tool. Authors also built 18 specialized polarity lexicons for both colloquial Arabic and MSA. Considered polarities are: positive, negative, and neutral. Experimental results showed that CNSA-MSA-SAT tool obtained an accuracy rate of 90% over the test dataset.

Al-Kabi et al. [22] developed an opinion mining and analysis tool to evaluate Arabic social content, both colloquial and MSA. Comments and reviews are evaluated according to three characteristics: subjectivity (subjective or objective), Polarity (positive or negative) and weight (strong or weak). Experimental results of the developed analysis tool showed that the proposed approach obtained more accurate results when it is applied on specific domain reviews (politics, technology, products, etc.).

Abdulla Nawaf et al. [23] have opted for a lexicon-based unsupervised sentiment analysis approach with a manual creation of the lexicon. Their approach is based on two components: the lexicon and the SA tool. The test dataset was created from two corpora each prepared separately. The first consists of 2000 tweets (1000 positive and 1000 negative) written in MSA and Jordanian dialect. The second corpus, extracted from Yahoo Maktoob, collected to meet the same criteria as the Twitter corpus (2000 comments with 1000 positive and 1000 negative). Authors obtained a low accuracy level in experiments, thus, they suggest some improvements by: expanding the lexicon, including the concept of weighted positive/negative sentiment for each word of the lexicon, proposing new improved combination techniques in the overall polarity computation, carrying out experimentations on bigger and diverse datasets, etc.

Hossam S. Ibrahim et al. [12] presented a feature-based sentence level sentiment analysis approach for Arabic language. They used a lexicon consisting of Arabic phrases to improve the polarity detection of Arabic sentences. Also, many linguistic features have been used, such as, Intensifiers, Shifters and negation. The developed lexicon focuses on both MSA and Egyptian dialectal Arabic. Experimental results showed that the proposed approach obtained 95% of accuracy using SVM classifier.

2.3 Related techniques to improve Arabic sentiment analysis

Shoukry and Rafea [11] studied the effect of pre-processing mechanisms on the performance of an Arabic sentiment analysis. Authors used a dataset consisting of 1000 tweets expressed in Egyptian Arabic dialect extracted from Twitter. They have used two stemmers over two approaches (Machine Learning (ML) and Sentiment Orientation (SO)). By using the pre-processing module combined with their stemmer, authors obtained improvement of 4.5% (respectively between 2-7%) for ML approach in all used measures (respectively for SO approach).

In [10], Shoukry and Rafea proposed an implementation of a sentiment classification for Arabic tweets. They investigated the use of the machine learning approach for Arabic sentence level sentiment analysis by using 1000 extracted tweets. Two Classifiers have been used: Naïve Bayes and SVM. Authors mentioned some problems related to the training corpus which could affect the classification accuracy.

Al-Kabi et al. [24] conducted a comparative study between two sentiment analysis tools, SocialMention and Twendz, by using a dataset containing 4,050 Arabic and English reviews collected from Yahoo news, YouTube, Facebook, Twitter, etc. Three polarity dictionaries (Arabic, English and emoticons) have been manually constructed based on this dataset. The experimental results has shown that SocialMention is more accurate to identify the polarity of Arabic/English comments compared to Twendz.

Salloum, Wael, and Nizar Habash [25] propose morphological analyzers for dialectal Arabic (called ADAM). They extend an MSA morphological analyzer's database through a set of handwritten rules to add new entries of dialectal affixes into this database. Experimental evaluation showed that ADAM has decreased to half the rate of out-of-vocabulary compared to SAMA.

Sadat et al. [26] developed a framework for Arabic dialects classification using probabilistic models across social media datasets. They carried out a set of experiments exploiting the n-gram technique with Markov language model and Naive Bayes classifiers. These experiments showed that Naive bayes classifier based on bi-gram model was able to get very good results by identifying 18 different Arabic dialects with an accuracy rate of 98%.

Saadane Houda and Nizar Habash [27] presented a basic layout of Algerian Arabic processing. This layout can be used in most of NLP applications, such as sentiment analysis. The authors carried out a comparison with other Arabic dialects (Egyptian, Tunisian, etc.).

3 Algerian Dialect

Algerian Arabic or Algerian dialect (ALGD) is considered as one of the most "hard to understand" Arabic dialects varieties. It is far less normalized and standardized compared to MSA. It has a vocabulary inspired from Arabic but the original words have been altered phonologically [28]. ALGD belongs to Maghrebi Arabic (Western group) and is mainly used in daily life. It is characterized by the absence of writing resources, hence it is considered as an under resourced language [27]. ALGD differs from MSA and other Arabic dialects by having many specific features. In addition to MSA and dialectal Arabic, a rich vocabulary consisting of foreign words of French origin are an essential part of the spoken language of Algerians.

Phonology, morphology, lexicon and syntax of ALGD are very difficult to understand for the citizens of the other Arab countries.

For historical reasons, ALGD has been enriched by many languages (Turkish, Italian, Spanish and mainly by French) which resulted a complex linguistic situation.

With the advent of social networks, the ALGD is increasingly used by the Algerian Web users. According to ITU[1], 28% of Algerians are actively using Internet. Most of this activity is dominated by using social networks. Millions of comments and reviews are added every day. Mining this enormous volume of comments and reviews

[1] International Telecommunication Union

requires taking into account particular aspects of ALGD. Thus, our use of ALGD will focus mainly on the written form and its characteristics.

The first feature of ALGD is the use of words that comes from several languages (Code-Switched). Algerian vernacular Arabic is often known as a dialect code-switched with French [29]. To illustrate this feature, we can give the example of a comment excerpted from our test corpus: "top 444 ربي يوفقك" in which the user has used the words "ربي" and "يوفقك" that are of Arab origin, and the words "top" and "444" (4 is expressed to represents the word fort which means strong) that are of French origin. This comment means "top, strong, god helps you ".

The second feature is related to the use of Arabic expressions encoded in Romanized Arabic or foreign expressions (mostly French) encoded in Arabic letters. As example of the first case (Arabic encoded Romanized, known also as *arabizi*), we mention the comment: " *itar kbir flblad w ..... ysab Addine Hadiya karitha* ", equivalent to the following expression in Arabic: "إطار كبير فالبلاد ويسب الدين.... هذي كارثة", which means: *"A senior executive of the state, and he insults the religion ... it's a disaster"*. For the second case (French encoded in Arabic letters), we mention the comment from our test corpus:"جامي نيتيليزي سكايب", that represents the French expression : "je n'utiliserai jamais skype", which means:" I will never use skype ".

The third feature is the combination of the two first features, i.e. code-switched expressions containing words encoded with Romanized Arabic mixed with French words (or other foreign languages) encoded in Arabic letters. We mention this example from our test corpus:"khorda عرة الطوموبيلات ". This expression contains an Arabic word of the Algerian dialect written in Arabic letter (عرة which means worst), a modified French word written in Arabic letters (الطوموبيلات which means cars) and a Romanized Arabic word (khorda which means scrap). This expression means "the worst car, scrap".

The last feature is related to the use of words written in a very specific form, the form that most Algerians generally used for writing short messages. For instance, the word "mli7" (which represents the Algerian Arabic word "مليح") which means "Good". Also, the word "3ayane" (which represents the Algerian Arabic word "عيان") which means in certain cases "tired" or "bad" in other cases. We note here the use of Arabic numerals to present Arabic letters that are close in their writing "7" for "ح", "3" for "ع", etc. and the use of abbreviations.

All these features make the spoken and written Algerian dialect a very rich and varied language which requires special consideration of all of these properties and linguistic diversity.

4 The Proposed Lexicon-Based Sentiment Analysis Approach

This section exhibits our proposed lexicon-based sentiment analysis approach. Our approach attempts to address several issues related to sentiment analysis for the specific case of Algerian vernacular Arabic. These problems mainly lie in:

- All features mentioned in section 3, related to specific aspects of ALGD;
- Unavailability of Algerian vernacular Arabic sentiment lexicons;

- Unavailability of Algerian vernacular Arabic parsers.
- Unavailability of test dataset for the Algerian vernacular Arabic content.

To address the first problem, i.e. related to specific aspects of ALGD mentioned in section 3, we propose a process to handle each of these aspects. Our process is based on four modules: common phrases similarity computation module; pre-processing module; language detection & stemming module; and polarity computation module.

To address the second problem, we built three lexicons which are: keywords lexicon (L1); negation words lexicon (L2); intensification-words Lexicon (L3). Two other resources were used: a list of emoticons with their assigned polarities, and a dictionary of common phrases of the ALGD.

Concerning the third point, namely Algerian vernacular Arabic parsers, we have implemented a parser which takes its strength from our knowledge of different forms of expression used by the Algerian Web users. Our parser is based on the three following steps: tokenization, normalization and stop-words removal.

To address the last point, i.e. test dataset problem, we collected "post and comments" during a period of about a month from several pages of the Facebook social network very well-known and frequented in most cases by Algerian Web users. These dataset has been filtered and annotated by experienced users to form our test corpus.

4.1 Algerian Vernacular Arabic Sentiment Lexicons

As aforementioned, our sentiment analysis approach is based on three lexicons:

- Keywords lexicon (L1);
- Negation words lexicon (L2);
- Intensification words Lexicon (L3).

To build our L1 lexicon, we relied on the work of the text mining research group at Nile University2 in which they set a lexicon containing the words and expressions in Arabic and Egyptian dialect annotated with their polarities. We firstly remove all words and expressions not used in the ALGD. After that, we have included all the words (with their respective polarities) of the ALGD equivalent to Arabic and Egyptian words. Finally, we added the words of the Algerian dialect commonly used to express positive or negative opinion.

At the end of these steps, our lexicon L1 was composed of 2380 words with a negative polarity and 713 with a positive polarity.

For the other two lexicons, we used an MSA dictionary of negation and intensification words. So, we added all equivalent words used in the ALGD to these lexicons.

As aforementioned, two other resources were prepared: a list of emoticons and a dictionary of common phrases of the ALGD with their assigned polarities.

[2] http://tmrg.nileu.edu.eg/

4.2 Lexicon-Based Sentiment Analysis Process for ALGD

Our lexicon-based sentiment analysis process is based on four modules:

- common phrases similarity computation module;
- pre-processing module;
- language detection & stemming module; and
- Polarity computation module.

Our approach may involve other work to better define the nature of expressions before processing. As an example we can cite the work of Sadat et al. [26] allowing the automatic identification of Arabic dialects.

Common phrases similarity computation module. The first module, i.e. common phrases similarity computation module, allows to deal with common expressions before passing at the word level handling. This module compare the input text (comment) with the "common phrases table" by computing its score of similarity (N-gram similarity). If the score of similarity exceeds a certain threshold, the module will consider the input text as a common phrase, therefore, no need to proceed to the word by word handling.

Pre-processing module. The pre-processing phase is very important for all NLP tasks. Shoukry and Rafea [11] indicated that this phase has a very positive impact on the performance of sentiment analysis.

The pre-processing module is mainly based on our parser, it extracts the tokens (keywords, negation words, intensification words and emoticons) by proceeding according to the following steps: tokenization, normalization and stop-words removal.

Arabic language is known by the property to have multiple forms of a given letter, for instance, "ا، أ، إ، آ، ا" are several forms for the letter "ا" (alif). Therefore, the normalization step serves to transform every letter to its defined standard form.

For the stop-words removal step, it consists often to remove common words that are unrelated to the topic of interest, such as "and" and "the" in English language. In the information retrieval (IR) field, this step is very important in both indexing and retrieval phases. All stop words are removed by the IR system. Contrary to this, some stop words can play an important role in SA field. For instance, we can cite the case of following stop words: "مع" and "و". The first word expresses an agreement, as used in the phrase: "أنا مع هذا الإقتراح", which means "I agree with this proposal". The second word, i.e "و", is used as a linking word between two sentences. In most of the cases, this word is used to link sentences referring to two aspects of the same topic. For instance, the expressions "هذا تلفون جميل و لكنه غالي", which means "this phone is beautiful but expensive ". For this, we have defined a list of stop words a bit limited compared to that used in the field of information retrieval.

The output of this module consists of a list of tokens (encoded in Arabic, Romanized or emoticons).

Language detection & stemming module. The processing performed by this module is detailed figure 1. It takes as input the results of the previous module. The first step of this diagram is to detect the language of a token T_i. If T_i belongs to Arabic, then the module will calculate its stem using a light stemming Arabic tool, for instance, khoja stemmer[3]. The other case is when T_i don't belongs to Arabic, we will have two sub-cases. The first sub-case is when a word belongs to another language. In this sub-case the module will carry out the translation of the word to Arabic encoded. As example, we can mention the word "formidable" that will be translated to "رائع". In the second sub-case, a specific translation is needed here, i.e. suggestion. For instance, the word "مليح" will be suggested by the translator (Google in our experiments) to replace the word "mlih" (Romanized Algerian Arabic word), which means "good".

At the last stage of this module, we will have as result a list of stemmed tokens.

Polarity computation module. This section describes (through the diagram of Fig. 2) the way our module computes the sentiment orientation of each term, and therefore aggregates these terms SO to obtain the SO of the entire text. After initialization of the text sentiment orientation (TSO) to 0, the first step of the polarity computation process consists in verifying the membership of the term T_i to the lexicon L1 (words with polarities, see section 4.1). The terms belonging to L1 are of three polarities (positive, neutral and negative). If the current term does not belong to L1, the module will process the next term of the text. Otherwise, a set of rules will be checked, primarily involving interaction with words belonging to the L2 and L3 lexicons.

The first rule attempts to verify if a term T_i is is preceded by a term T_k and succeeded by a term T_j, where T_k belongs to L2 and T_j belongs to L3. If this is the case, the value "$Weight(T_k)* SO(T_i)$" will be added to the *TSO*.

As example of this rule, we can mention the case of the phrase: "ماشي مليح بزاف", which means "it is not so good".

As example of the second rule, we can mention the subexpression "مليح بزاف", which means "So good".

The third rule processed in this module is related to the negation case. For instance, the subexpression "ماشي مليح", which means "Not good". If this rule is verified, the module will add the value "$Weight(T_k)* SO(T_i)$" to the *TSO*. If not, only the SO of the term T_i will be added to *TSO*.

4.3 Test Corpus

In this section, we will explore the different characteristics of our test corpus. This corpus will be used in the experimental phase of the present research. This step consisting in collecting and annotating the dataset (assigning a polarity to each comment) is very expensive in terms of both time and effort.

[3] See: http://zeus.cs.pacificu.edu/shereen/research.htm

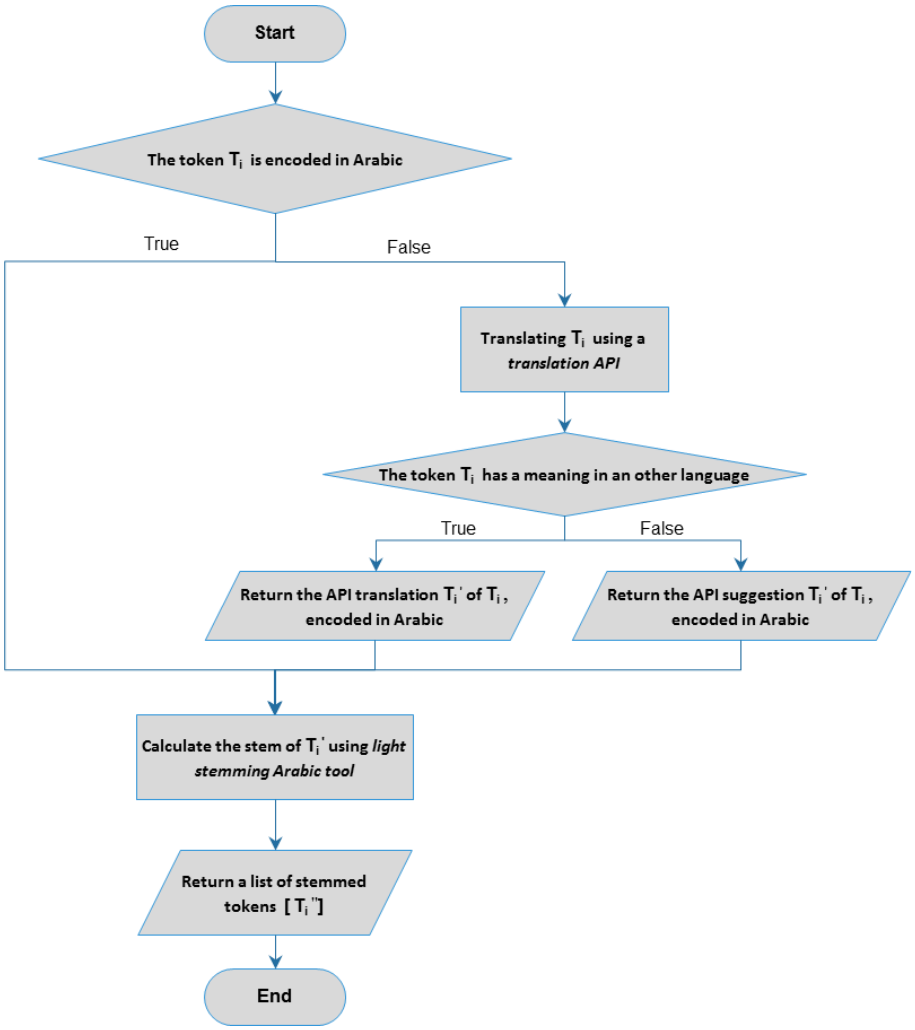


Fig. 1. Processing diagram of the language detection & stemming module

Data collection. Our data (post and comments) are exclusively extracted from Facebook. This is justified by the high use of this social network by Algerian Web users (more than 96%[4]). According to Facebook, 11M Algerians use this social network.

We have developed a module based on Facebook4J[5] for the data extraction from the Facebook social network.

[4] http://gs.statcounter.com/#social_media-DZ-monthly-201501-201601-bar
[5] http://facebook4j.org/

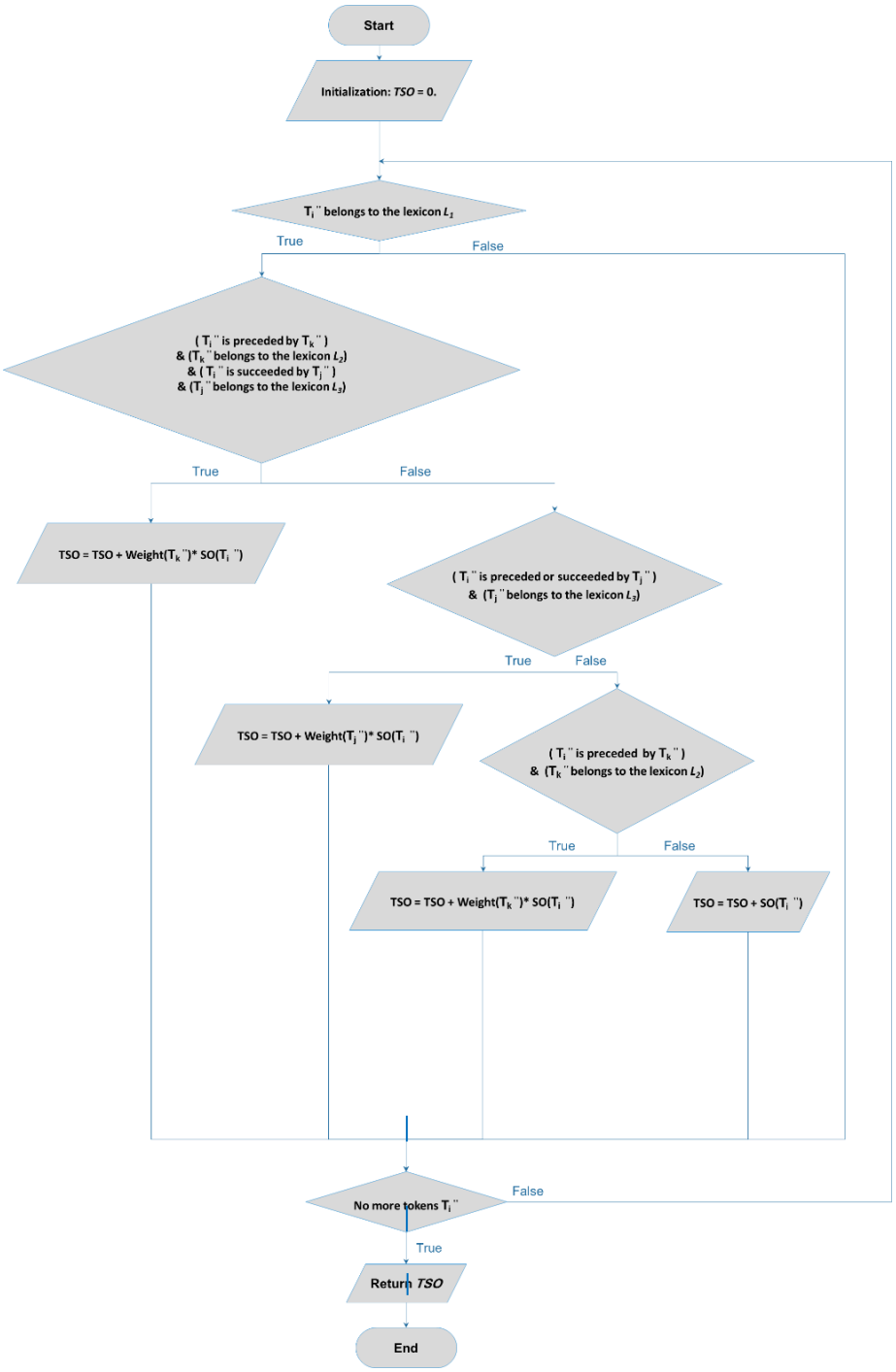


Fig. 2. Processing diagram of the polarity computation module

In order to target the right content, we made use of data provided by socialbakers[6]. These data show the main frequented Algerian pages. Thus, we have chosen the following pages:

- "احلام مستغانمى", a writer with more than 9,154,070 fans;
- "خديجة بن قنة", a journalist with more than 7,430,493 fans;
- "Lotfi DK", a rap songer with more than 4,409,397 fans;
- "Ooredoo", a telecom operator with more than 2,972,330 fans.

etc. These pages discuss various areas of life, i.e. economy, politics, literature and arts, etc.

The collected data has been filtered and annotated by experienced users to form our test corpus.

Dataset Properties. In total, we selected 206 posts comprising 7698 comments. . As aforementioned, all these comments were assessed manually by defining their polarities. The collected data discuss various areas of life from which we can mention the pages: economy ("Ooredoo", "Djezzy", "Mobilis", "Samsung Algérie"), politics ("Ali Benflis", "عبد العزيز بوتفليقة"), society ("Algérie"), literature and arts ("احلام مستغانمى", "خديجة بن قنة", "DZjoker"), sports ("الفريق الوطني الجزائري", " Maracana"), divers ("Hafid Derradji", "Journal el Bilad", "Zinou Kds", "fibladi.com", "Karim El Gang", "Echorouk online", "1.2.3 viva l'algerie", "El khabar"), etc. Table 1 shows the distribution of the collected data according to their topics.

Table 1. Distribution of the collected data according to their topics

Topic	number of posts	number of comments
economy	68	1705
politics	33	2422
society	32	1263
literature and arts	49	1215
divers	24	1093

As it can be seen from Table 1, the posts belonging to politics and society topics has the highest frequency of comments.

In Table 2, we present data from a perspective of the lexicon and encoding used. We note that most of comments (56%) use the ALGD encoded in Arabic or Romanized. Foreign comments are mainly words from the French.

Table 2. Number of comments according to thier Encoding class

	Number of comments
MSA encoded with Arabic letters	1503
MSA encoded with Romanized letters	36
ALGD encoded with Arabic letters	2429

[6] http://www.socialbakers.com/statistics/facebook/pages/total/algeria/

ALGD encoded with Romanized letters	1909
Foreign languages encoded with Romanized letters	1281
Foreign languages encoded with Arabic letters	7
MSA and/or ALGD encoded with Arabic letters	312
MSA and/or ALGD and/or Foreign languages encoded with Arabic letters and/or Romanized letters	221

We present in the following table some examples of comments from our test corpus according to their encoding classes (aforementioned in Table 2).

Table 3. Examples of comments from our test corpus

Original comment	Translated Comment	Encoding class
والله صحيح, شكرا أحلام على النصيحة	Right, Thanks Ahlam for the advice	MSA encoded with Arabic letters
Sabah elward, kalam jamil	Good morning, nice words	MSA encoded with Romanized letters
راك فور خو	You are strong brother	ALGD encoded with Arabic letters
ma 3andehomch anti derapage hhhh	they cannot change their minds	ALGD encoded with Romanized letters
Facebook devrait créer le bouton "J`adore" Lotfi DK.	Facebook should create the button "I love" Lotfi DK.	Foreign languages encoded with Romanized letters
تووووووب	Top	Foreign languages encoded with Arabic letters
بالتوفيق و لعقوبة لنجاحات أخرى .	Good luck, And other successes.	MSA and/or ALGD encoded with Arabic letters
bonne chance rabi m3akoum	Good luck, god helps you	MSA and/or ALGD and/or Foreign languages encoded with Arabic letters and/or Romanized letters

4.4 Experimental Results

In this section we present the obtained experimental results. Experiments were conducted based on our constructed ALGD dataset by using classic precision measure (accuracy).

From results of table 4, we can observe that the best configuration of our experiments is related to the use of the combination: "Arabization + Translation + khoja Stemmer".

To test the impact of the "common phrases similarity computation module", we have defined two configurations (with and without this module). According to table 5, the obtained results show that this module allowed us to improve the accuracy of our system.

Table 4. Impact of arabization, translation and stemming phases

	Basic analyzer	with Arabization	with Arabization + Translation	with Arabization + Translation + Light Stemmer	with Arabization + Translation + Khoja Stemmer
Accuracy	53.3%	65.0 %	71.9 %	72.05%	76.68 %

Table 5. Results obtained by the two configurations related to the "common phrases similarity computation module"

	Without using "common phrases similarity computation module"	By using "common phrases similarity computation module"
Accuracy	76.68 %	79.13 %

5 Conclusion

We proposed in this paper a new lexicon-based approach for vernacular Algerian Arabic sentiment analysis. This approach attempts to address the specific aspects of this very particular Arabic dialect. All these aspects that were apparent before that in spoken language, but now with the advent of social networks these features exist throughout the generated content of Algerian Web users.

We mentioned in this work the main issues related to these features and proposed an approach composed of four modules: common phrases similarity computation module; pre-processing module; language detection & stemming module; and polarity computation module. Our built lexicon is composed of three parts: keywords lexicon; negation words lexicon; intensification words lexicon. These three lexicons are enriched by a dictionary of emoticons and another dictionary of common phrases.

Finally, we have built a test corpus for experimental purposes. This corpus was filtered and annotated in order to facilitate the evaluation process of our proposal.

Experimental results show that our system obtains good performance with 79.13% of accuracy.

Acknowledgment

We thank all members of the "*Text mining research group*" of *Nile University* for giving us the opportunity to exploit their data sets.

References

1. Al-Kabi, M., et al. *An opinion analysis tool for colloquial and standard Arabic*. in *The fourth International Conference on Information and Communication Systems (ICICS 2013)*. 2013.
2. Pang, B. and L. Lee, *Opinion mining and sentiment analysis.* Foundations and trends in information retrieval, 2008. **2**(1-2): p. 1-135.
3. He, Y. and D. Zhou, *Self-training from labeled features for sentiment analysis.* Information Processing & Management, 2011. **47**(4): p. 606-616.
4. Taboada, M., et al., *Lexicon-based methods for sentiment analysis.* Computational linguistics, 2011. **37**(2): p. 267-307.
5. Ding, X., B. Liu, and P.S. Yu. *A holistic lexicon-based approach to opinion mining*. in *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 2008. ACM.
6. Kumar, A. and T.M. Sebastian, *Sentiment analysis on twitter.* IJCSI International Journal of Computer Science Issues, 2012. **9**(3): p. 372-378.
7. Klenner, M., S. Petrakis, and A. Fahrni. *Robust Compositional Polarity Classification*. in *RANLP*. 2009.
8. Pak, A. and P. Paroubek. *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*. in *LREC*. 2010.
9. Duwairi, R.M., et al. *Sentiment Analysis in Arabic Tweets*. in *Information and Communication Systems (ICICS), 2014 5th International Conference on*. 2014. IEEE.
10. Shoukry, A.M., *Arabic sentence level sentiment analysis*. 2013, The American University in Cairo.
11. Shoukry, A. and A. Rafea. *Preprocessing Egyptian Dialect Tweets for Sentiment Mining*. in *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages*. 2012.
12. Ibrahim, H.S., S.M. Abdou, and M. Gheith, *Sentiment Analysis For Modern Standard Arabic And Colloquial.* arXiv preprint arXiv:1505.03105, 2015.
13. Abdulla, N.A., M. Al-Ayyoub, and M.N. Al-Kabi, *An extended analytical study of arabic sentiments.* International Journal of Big Data Intelligence 1, 2014. **1**(1-2): p. 103-113.
14. Elkhlifi, A. and R. Bouchlaghem, *Opinion Extraction in Moroccan Dialect Texts.*
15. Abdul-Mageed, M. and M.T. Diab. *SANA: A Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis*. in *LREC*. 2014.
16. Badaro, G., et al., *A large scale Arabic sentiment lexicon for Arabic opinion mining.* ANLP 2014, 2014: p. 165.
17. Diab, M., et al. *Tharwa: A large scale dialectal arabic-standard arabic-english lexicon*. in *Proceedings of the Language Resources and Evaluation Conference (LREC)*. 2014.
18. Al-Kabi, M., et al., *A Prototype for a Standard Arabic Sentiment Analysis Corpus*, in *The International Arab Conference on Information Technology (ACIT'2015)*. 2015: Amman, Jordan.
19. Itani, M.M., et al. *Classifying sentiment in arabic social networks: Naïve search versus Naïve bayes*. in *Advances in Computational Tools for Engineering Applications (ACTEA), 2012 2nd International Conference on*. 2012. IEEE.

20. Ahmad, K., D. Cheng, and Y. Almas. *Multi-lingual sentiment analysis of financial news streams*. in *Proc. of the 1st Intl. Conf. on Grid in Finance*. 2006.
21. El-Beltagy, S.R. and A. Ali. *Open issues in the sentiment analysis of Arabic social media: A case study*. in *Innovations in information technology (iit), 2013 9th international conference on*. 2013. IEEE.
22. Al-Kabi, M.N., et al., *Opinion mining and analysis for arabic language.* IJACSA) International Journal of Advanced Computer Science and Applications, 2014. **5**(5): p. 181-195.
23. Abdulla, N.A., et al., *Towards improving the lexicon-based approach for arabic sentiment analysis.* International Journal of Information Technology and Web Engineering (IJITWE), 2014. **9**(3): p. 55-71.
24. Al-Kabi, M., et al. *Arabic/English sentiment analysis: an empirical study*. in *The Fourth International Conference on Information and Communication Systems (ICICS 2013)*. 2013.
25. Salloum, W. and N. Habash, *ADAM: Analyzer for Dialectal Arabic Morphology.* Journal of King Saud University-Computer and Information Sciences, 2014. **26**(4): p. 372-378.
26. Sadat, F., F. Kazemi, and A. Farzindar. *Automatic identification of Arabic dialects in social media*. in *Proceedings of the first international workshop on Social media retrieval and analysis*. 2014. ACM.
27. Saadane, H. and N. Habash. *A Conventional Orthography for Algerian Arabic*. in *ANLP Workshop 2015*. 2015.
28. Meftouh, K., N. Bouchemal, and K. Smaïli. *A study of a non-resourced language: an Algerian dialect*. in *SLTU*. 2012.
29. Cotterell, R., et al. *An algerian arabic-french code-switched corpus*. in *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme*. 2014.

A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis

Heba M. Ismail[1], Saad Harous[1], Boumediene Belkhouche[1]
[1] College of Information Technology,
United Arab Emirates University, Al Ain, UAE
{hebaismail20@gmail.com},{harous, b.belkhouche}@uaeu.ac.ae

Abstract. Twitter popularity has increasingly grown in the last few years making influence on the social, political and business aspects of life. Therefore, sentiment analysis research has put special focus on Twitter. Tweet data have many peculiarities relevant to the use of informal language, slogans, and special characters. Furthermore, training machine learning classifiers from tweets data often faces the data sparsity problem primarily due to the large variety of Tweets expressed in only 140-character. In this work, we evaluate the performance of various classifiers commonly used in sentiment analysis to show their effectiveness in sentiment mining of Twitter data under different experimental setups. For the purpose of the study the Stanford Testing Sentiment dataset STS is used. Results of our analysis show that multinomial Naïve Bayes outperforms other classifiers in Twitter sentiment analysis and is less affected by data sparsity.

1. Introduction

The use of social networking websites such as Twitter and Facebook has been witnessing a rapid growth in the last few years. Probably the reason behind this increase is that people feel comfortable expressing their views and opinions casually on a wide array of topics via such websites. On the other hand, our decision-making process is oftentimes influenced by other people's opinions. Most of us would seek our friends', family members', or co-workers' recommendations before making important purchase decisions, before eating at a specific restaurant, or watching a new movie. Sometimes we even base our decision solely on those opinions. To this end, sentiment analysis has attracted a huge research interest especially in recent years. Researchers analyzed sentiment in many domains: movie reviews, news articles, blogs, forums, product reviews, and more recently social media data. Sentiment analysis of data available on the social networks which comprises of people's views is becoming very important in order to gauge public opinion on a particular topic of interest. It can help evaluate consumer satisfaction about some products, customers' interests and preferences, political viewpoints and many others. Indeed, number of surveys shows that:

- 91% of people visited a store because of an online experience. Among which 22% were influenced by Twitter and Facebook experiences[1]
- 72% of consumers trust online reviews as much as personal recommendations[2]
- 78% of consumers state that posts made by companies on social media influence their purchases[3]

Twitter amongst other social networks is becoming the most popular and influential social network. Every month, millions of people tweet about what they love: products they buy, places they visit, books they are reading, vacations they are planning, and public figures or politicians they like or dislike. Such an enormous amount of public opinions can be of great value. As well as, it can be challenging to identify and engage with the most relevant Tweets about specific topic of interest at the time they are needed. In order to reveal the overall sentiment of the population, retrieval of data from such sources and subsequent sentiment analysis becomes vital.

Sentiment analysis on text is a very difficult task by itself, given the unstructured or in the best cases ill-structured nature of text along with the context complexity [4], let alone extracting sentiment from a text as noisy as social media text. There are some difficulties inherent in analyzing sentiment from social media [5]. One example is "False negatives" where words such as "crying" and "crap" generally suggest negativity, yet they imply positive sentiment when used in a sentence such as "I was crying with joy" or "Holy crap! This is great". Another example is "Conditional sentiment," such as "If someone doesn't call me back, I will never do business with them again." These examples show how sentiment analysis of social media text can be hard. Moreover, the process gets even harder with the use of emoticons such as "." ("smiley) and hash-tags such as "#happy" to express feelings ironically or sarcastically. In addition to the previous and in particular to Twitter, text is usually very short, whereby a maximum Tweet size is 140 characters, and as a consequence, the generated dataset for a specific Twitter corpus may have very large feature space with few values for each Tweet, resulting in a highly sparse dataset that negatively influences the accuracy of the sentiment analysis. These inherent problems in social media text in general and in Twitter in particular impose significant challenges on the sentiment analysis process.

Machine learning classifiers have been widely used for the purpose of sentiment mining providing good accuracy results. Different research studies, reported different accuracy results for unigrams (i.e. distinct words in the corpus) vs bigrams (i.e. combination of every two consecutive words in the text). As well as, different accuracy results were reported for using term frequency vs term presence in the document.

Yet, there is no formal empirical study evaluating the effect of different input representation on the performance of the classifiers. Hence, our study analyzes formally the performance of sentiment classification methods based on fair experimental setups. We analyze unigrams, as well as, bigrams as features spaces. For example for a tweet "I Love Kindle, It's Amazing", unigrams = {I, Love, Kindle, Its, Amazing}, bigrams = {I Love, Love Kindle, Kindle Its, Its Amazing}. Moreover, we analyze term frequency representation of dataset (i.e. the number of occurrences of a term in a document), as well as, term presence representation (i.e. the occurrence or absence of a term in a document regardless of how many times it occurred). For training and testing we are using Stanford Testing Dataset. Details about the experimental setups are provided in section VI. In the following section we review some related works to ours then we present a brief overview of sentiment analysis and highlight the major areas of research in sentiment analysis. Section IV presents some commonly used classifiers in sentiment analysis. Finally we present our experiment setup and results.

2. Related Work

In the literature there are few studies that attempted to empirically evaluate the performance of classification algorithms in sentiment mining. Vinodhini and Chandrasekaran [6] conducted a comparative study on four classifiers: K-Nearest Neighbors, Decision Trees, Naïve Bayes and Support Vector Machines, to evaluate their performance in sentiment mining of online product reviews. They used different sampling methods (e.g. linear sampling, bootstrap sampling and random sampling) to create training examples from the product reviews dataset. Their results show that support vector machine with bootstrap sampling method outperforms other classifiers and sampling methods in terms of misclassification rate. They used unigrams for feature space and terms occurrences to populate the classification input. They did not provide any information about the influence of input format on the classification results.

On the other hand, Hang et al. [7] evaluated the performance of three classifiers: Passive-Aggressive (PA) Algorithm Based Classifier, Language Modeling (LM) Based Classifier and Winnow, using 100K online product reviews with focus on the impact of higher order n-grams ($n > 3$). They found that discriminating classifier (i.e. Passive-Aggressive Based) combined with high order n-grams as features can achieve comparable, or better performance than that reported in academic papers. Hang et al study analyzed up to 6-grams feature length. However, they did not show the impact of data representation (e.g. frequency, occurrences) on the performance of the classifiers.

Furthermore, Vinodhini's and Hang's studies were conducted on product reviews which may have length up to 800 characters or more. However, a Tweet is limited to 140 character which adds another challenge to sentiment mining of Twitter. Normally datasets generated from Twitter suffer from large sparsity. Higher order n-grams may not be suitable to use as a Tweet may have 6, 5 or 4 words only which are used by Hang's as n-grams features. Given the special peculiarities of twitter text and length, a twitter-specific comparative study is needed to evaluate the performance of popular classification algorithms in the area of sentiment mining using different input formats which actually have the direct impact on the classification accuracy.

3. Sentiment Analysis

Sentiment mining, polarity mining, opinion mining or sentiment analysis is concerned with analysis of direction-based text, i.e. text containing opinions and emotions. Sentiment analysis involves many tasks. Four of the important tasks of sentiment analysis where most of the research effort is focused are: data preprocessing, class labeling, annotation granularity, and target identification [8]. Data preprocessing is vital especially for the text collected from social media websites because it is unstructured and full of spelling mistakes and peculiarities. All researchers in the area of sentiment analysis perform some or all of the natural language preprocessing tasks including: spellchecking, and stop words removal such as punctuation marks. In addition, some researchers perform stemming before classification [9] [10]. In class labeling process (i.e. the process of annotating text into labels or classes) some

research focuses on categorizing text as subjective or objective. In sentiment analysis, this task is usually carried out first, because it was proven that performing it prior to polarity classification improves the latter [4]. In other words, if a text is identified as subjective then we can perform polarity classification to determine whether this subjective text is carrying positive sentiment or negative sentiment. On the other hand, a large body of research focuses on automating the process of class labeling through distant supervision using noisy labels.

For example, [11] used emoticons such as ":-)" and ":(" to label tweets as positive or negative. However, [12] argued that using noisy sentiment labels may hinder the performance of sentiment classifiers. They proposed exploiting the Twitter follower graph to improve sentiment classification and constructed a graph that has users, tweets, word unigrams, word bigrams, hashtags, and emoticons as its nodes which are connected based on the link existence among them (e.g., users are connected to tweets they created; tweets are connected to word unigrams that they contain etc.). Then they applied a label propagation method where sentiment labels were propagated from a small set of nodes seeded with some initial label information throughout the graph. Having a pre-processed subjective text with class labels, sentiment classification can be conducted at the document [13], sentence [14] or phrase levels [15] (where a phrase is part of a sentence) which we refer to as the granularity of the classification. Finally, knowing the source and the target of a sentiment is considered as one of the challenges of sentiment analysis that was addressed by number of researchers [16].

4. Machine Learning Classifiers For Sentiment Analysis

The two most commonly used approaches in sentiment analysis techniques are: the lexicon-based approach and the learning approach [17]. Lexicon based approaches are used widely to classify unsupervised text sentiment. Such classifiers attempt to classify data on the number of positive and negative words present in the text, and do not need any training dataset. These words which express opinion are known as "opinion words" and the lexicon is known as "opinion lexicon". Basically in the lexicon based approaches we rely on external lexical resources that associate polarity score to each term. Sentiment of text depends on the sentiment of the terms that compose it. Examples of lexical dictionaries are: (i) SentiWordNet, (ii) WordNet Affect, (iii) Sentic Net and (iv) MPQA. The major problem with this approach is that there is no mechanism to deal with context dependent words. For example, the word," Long" can be used to convey a positive as well as a negative opinion both depending upon the context in which it is used. For example, we can think of two sentences as "This mobile takes long time to charge" which is a negative opinion, whereas saying "This mobile phone has long battery life" is a positive opinion. On the other hand, classification approaches involve building classifiers from labeled instances of texts or sentences, essentially a supervised classification task. In our research we focus on classification learning approaches for sentiment analysis. In the following sections we explore some of the most commonly used machine learning classifiers for sentiment analysis

4.1. Naïve Bayes Classifiers

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem. The descriptive attributes/features are assumed to be conditionally independent from each other, which makes a naïve assumption [18]. Typically, due to the independence assumption, the class-conditional probability for an object X (i.e. which is a record or a row in the dataset), is estimated as the product of all independent events' (i.e. Features' Values, X1, X2, X3 …. Xd) conditional probabilities for a given class Y, such that:

$$P(X \mid Y = y) = \prod_{i=1}^{d} P(Xi|Y=y)$$

Therefore, for predicting a class Y:

$$P(Y = y \mid X) = P(Y = y) \left(\prod_{i=1}^{d} P(Xi|Y=y) \right) / P(X)$$

Since P(X) is a common denominator for all class prediction calculations for a single record (X), it does not affect the choice of the class; therefore we can replace the previous formula with the following:

$$P(Y = y \mid X) = P(Y = y) \left(\prod_{i=1}^{d} P(Xi|Y=y) \right)$$

Major strengths of naïve Bayes classifier are: handling noisy data since it is averaged out in the estimation of conditional probability, null values are ignored and irrelevant features are uniformly distributed so they do not have significant influence on the classification result. Weaknesses are mainly attributed to the assumption of complete independence amongst attributes. If there are no occurrences of a class label and a certain attribute value together (e.g. class="nice", shape="sphere") then the frequency-based probability estimate will be zero. Given Naive-Bayes' conditional independence assumption, when all the probabilities are multiplied we will get zero and this will affect the posterior probability estimate. This problem happens when we are drawing samples from a population and the drawn vectors are not fully representative of the population. Lagrange correction and other schemes have been proposed to avoid this undesirable situation. There are several Naive Bayes variations. Here we will consider two of them: the Multinomial Naive Bayes, and the Bernoulli Naïve Bayes

4.2 Multinomial Naïve Bayes Text Classifiers

Using the Multinomial Naïve Bayes Text Classifier, the probability of a document d being in class c is computed as [19]:

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

where P(tk|c) is the conditional probability of term tk occurring in a document of class c. We interpret P(tk|c) as a measure of how much evidence tk contributes that c is the correct class. P(c) is the prior probability of a document occurring in class c. If a

document's term does not provide clear evidence for one class versus another, we choose the one that has a higher prior probability. (t1, t2, . . . , tnd) are the tokens in d that are part of the vocabulary we use for classification and nd is the number of such tokens in d. For example, (t1, t2, . . . , tnd) for the one-sentence document "Beijing and Taipei join the WTO" might be (Beijing, Taipei, join, WTO), with nd = 4, if we treat the term "and" as a stop word. In text classification, our goal is to find the best class for the document. The best class in Naïve Bayes classification is the most likely or maximum posteriori (MAP) class cmap:

$$c_{\text{map}} = \arg\max_{c \in \mathbf{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbf{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c).$$

P^(c) is calculated by finding the frequency of class c relative to the total size of the given training data such that:

$$\hat{P}(c) = \frac{N_c}{N},$$

where Nc is the number of documents in class c and N is the total number of documents. P^(tk|c) is calculated by finding the number of occurrences of t in training documents from class c, including multiple occurrences of a term in a document such that:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}},$$

where Tct is the number of occurrences of t in training documents from class c, including multiple occurrences of a term in a document. In the implementation of Multinomial Naïve Bayes (MNB) we need to add a smoothing one to the conditional probability so as to avoid zero probability of new terms in the testing set that were not available in the training set:

From the pseudocode in figure 1, we see that the complexity of the training process is Θ (|C||V|) because the set of parameters consists of |C||V| conditional probabilities and |C| priors. The time complexity for text pre-processing will be Θ (|D| * Lavg) (i.e. number of documents times the average length of documents) [19]. In this study we do not consider the pre-processing time since we are using the classifier as an independent program not including text pre-processing.

4.3 Bernoulli Naïve Bayes Text Classifiers

An alternative to the multinomial model is the multivariate Bernoulli model or Bernoulli model, which generates an indicator for each term of the vocabulary, either 1 indicating presence of the term in the document or 0 indicating absence. Figure 2 shows the training and testing algorithms for the Bernoulli model. This model estimates P^(t|c) as the fraction of documents of class c that contain term t. The Bernoulli model has the same time complexity as the multinomial model.

```
TRAINMULTINOMIALNB(C, D)
 1  V ← EXTRACTVOCABULARY(D)
 2  N ← COUNTDOCS(D)
 3  for each c ∈ C
 4  do N_c ← COUNTDOCSINCLASS(D, c)
 5     prior[c] ← N_c / N
 6     text_c ← CONCATENATETEXTOFALLDOCSINCLASS(D, c)
 7     for each t ∈ V
 8     do T_ct ← COUNTTOKENSOFTERM(text_c, t)
 9     for each t ∈ V
10     do condprob[t][c] ← (T_ct + 1) / Σ_t'(T_ct' + 1)
11  return V, prior, condprob

APPLYMULTINOMIALNB(C, V, prior, condprob, d)
1  W ← EXTRACTTOKENSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6  return arg max_c∈C score[c]
```

Figure 1. Naïve Bayes Multinomial Algorithm [19]

```
TRAINBERNOULLINB(C, D)
1  V ← EXTRACTVOCABULARY(D)
2  N ← COUNTDOCS(D)
3  for each c ∈ C
4  do N_c ← COUNTDOCSINCLASS(D, c)
5     prior[c] ← N_c / N
6     for each t ∈ V
7     do N_ct ← COUNTDOCSINCLASSCONTAININGTERM(D, c, t)
8        condprob[t][c] ← (N_ct + 1)/(N_c + 2)
9  return V, prior, condprob

APPLYBERNOULLINB(C, V, prior, condprob, d)
1  V_d ← EXTRACTTERMSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ V
5     do if t ∈ V_d
6           then score[c] += log condprob[t][c]
7           else score[c] += log(1 − condprob[t][c])
8  return arg max_c∈C score[c]
```

Figure 2. Bernoulli Naive Bayes Algorithm [19]

4.4. Support Vector Machines Classifiers

Support Vector Machine (SVM) is a non-probabilistic binary linear classifier that constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks. The main idea underlying SVM for sentiment classification is to find a hyper plane which divides the documents, or in our case, tweets as per the sentiment, and the margin between the classes being as high as possible [17]. For example, if we have a training set expressed mathematically as follows:

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$$

where xi is an n-dimensional real vector (i.e. document or a tweet in our case), yi is either 1 or -1 denoting the class to which the point xi belongs. First, The SVM classification function F(x), must return positive numbers for positive data points and negative numbers otherwise, that is, for every point xi in D. Second, F(x) (or the hyperplane) needs to maximize the margin. The margin is the distance from the hyperplane to the closest data points or vector (i.e. which will be called the support vector). This turns the SVM classifier into an optimization constraint problem. Solving this problem using Lagrange multipliers, the solution can be written as [20]:

$$F(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} - b$$

where the auxiliary nonnegative variable α is called Lagrange multipliers, and b is the bias, which will be computed by the SVM in the training process. Note that according to the property of Kuhn–Tucker conditions of optimization theory, the solution of the dual problem α must satisfy the following condition:

$$\alpha_i^* \{y_i(\mathbf{w}^* \cdot \mathbf{x}_i - b) - 1\} = 0 \quad \text{for } i = 1, 2, \ldots, m$$

and either α or its corresponding constraint {yi (w.xi – b) -1} must be nonzero. This condition implies that only when xi is a support vector or {yi (w.xi – b) -1} = 1, its corresponding coefficient αi will be nonzero (or nonnegative). After exploring the theoretical background of SVM, we understand that it is not an algorithm, but rather a mathematical relationship which leads to an optimization problem. This problem obviously requires an optimization algorithm to solve.

4.5 Sequential Minimal Optimization

Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines. SMO is widely used for training support vector machines and is implemented by popular data mining tools such as Weka.

SMO breaks the SVM optimization problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers αi, the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2, the constraints are reduced t_0:

$$0 \leq \alpha_1, \alpha_2 \leq C,$$

$$y_1\alpha_1 + y_2\alpha_2 = k,$$

where C is an SVM hyper-parameter, and k is the negative of the sum over the rest of terms in the equality constraint, which is fixed in each iteration. This reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. The training algorithm proceeds as follows:

- Find a Lagrange multiplier α1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem
- Pick a second multiplier α2 and optimize the pair (α1, α2),
- Repeat steps 1 and 2 until convergence

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem is then solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence. This is critical for large data sets since there are n*(n − 1) possible choices for αi and αj. In the worst case the algorithm has complexity of Θ (n3), where n is the number of vectors.

5. Proposed Method

In this comparative study we need to evaluate the performance of Multinomial NB, Bernoulli NB and SVM in sentiment mining of Twitter data. The selected classifiers are the most commonly used machine learning classifiers in the literature [11], [12], [17], [21], [22]. For comparison we use a selected Twitter dataset, apply suitable preprocessing steps then produce the dataset with unigrams and bigrams, one time with term frequencies and one time with term presence (i.e. polarity dataset). Afterwards, the three selected classifiers are trained with the four variations of the input dataset and the accuracy results are compared along with training time.

6. Experimental Setup

Our experimental setup is as follows.

6.1. Dataset

In the work conducted in this paper, we use the Stanford Twitter Sentiment Data which was collected between the 6th of April and the 25th of June 2009 [11]. The original test set consists of 177 negative and182 positive manually annotated tweets.

6.2. Pre-Processing and Feature Reduction

Natural language processing of the corpus is performed for stop words removal, bag of words extraction and equivalence classes' replacement such that:

- All Twitter usernames, which start with @ symbol, are replaced with the term "USERNAME".
- All URL links in the corpus are replaced with the term "URL"
- Reduce the number of letters that are repeated more than twice in all words. For example the word "haaaappy" becomes "haappy" after reduction
- Remove all Twitter hashtags which start with the#.
- Remove all emoticons as they add noise during the training of the classifiers

We choose unigrams (i.e. distinct words in the corpus), as well as, bigrams (i.e. combination of every two consecutive words in the text) as features spaces. For example for a tweet "I Love Kindle, It's Amazing", Unigrams would be {I, Love, Kindle, Its, Amazing}, whereas bigrams would be {I Love, Love Kindle, Kindle Its, Its Amazing}. Consequently, bigrams normally produces larger feature space.

6.3. Performance Evaluation Steps

After pre-processing is done, four different variations of the input dataset are produced:

- Unigram with term polarity.
- Unigram with Term Frequency.
- Bigrams with term polarity.
- Bigrams with term Frequency.

We choose Weka for evaluating the performance of the selected classifiers as it has exactly similar implementation to the one discussed in this paper for MNB, BNB and SMO. For one iteration, the dataset inputs are used to train the classifiers and results are verified using 10-fold cross validation. For the second iteration, only 66% of the dataset is used for training whereas the remaining is used for testing.

7. Results and Discussion

Table 1 shows the experimental classification results for Bernoulli Naïve Bayes, Multinomial Naïve Bayes and SOM classifiers with 10-fold cross validation. The results show that overall accuracy for unigrams datasets are higher than the accuracy for bigrams datasets. Furthermore, training time for unigrams dataset is in general less than bigrams. This is expected since bigrams produce larger feature space. Multinomial Naïve Bayes produced the best classification results with frequency, unigrams dataset. SOM requires the longest training time to build the model and does not outperform other classifiers in the context of sentiment analysis of Twitter, which makes it less preferable choice for sentiment analysis compared to multinomial NB that produces good accuracy results at very high training speed.

Table 2 shows the experimental classification results for Bernoulli Naïve Bayes, Multinomial Naïve Bayes and SOM classifiers with STS set divided into training set and testing set. The first observation is that training time did not significantly change. This means that using either method, cross validation or training set would take comparable training time for model building. For MNB and BNB the classification

results of bigrams with training outperformed the classification results with cross validation. For Bernoulli NB the accuracy of unigrams dropped compared to significant increase in the performance of SOM. However, overall accuracy results for unigrams still outperform bigrams. This is expected as well, since bigrams produce datasets that are sparser given the limit of 140 character of Twitter. Moreover, multinomial NB still outperforms other classifiers. We deduce that MNB is less affected by the data sparsity problem inherent in Twitter datasets.

Table 1. Testing Results -10-Fold Cross Validation

Classifier	Dataset Type	Unigrams (1442 Features)	Time (Sec)	Bigrams (4150 Features)	Time (Sec)
BNB	**Polarity**	**76.6%**	**0.22**	**70.75 %**	**0.61**
	Frequency	**75.21 %**	**0.24**	**65.18 %**	**0.61**
MNB	**Polarity**	**79.39 %**	**0.13**	**75.77 %**	**0.13**
	Frequency	**81.34 %**	**0.05**	**72.14 %**	**0.11**
SVM	**Polarity**	**74.37 %**	**4.34**	**74.09 %**	**12.16**
	Frequency	**77.16 %**	**4.22**	**69.95 %**	**12.95**

Table 2. Testing Results - Dataset split into 66% training set Conclusion

Classifier	Dataset Type	Unigrams (1442 Features)	Time (Sec)	Bigrams (4150 Features)	Time (Sec)
BNB	**Polarity**	**73.76%**	**0.22**	**73.77 %**	**0.59**
	Frequency	**73.77 %**	**0.25**	**68.03%**	**0.59**
MNB	**Polarity**	**82.78%**	**0.13**	**80.32 %**	**0.11**
	Frequency	**80.32 %**	**0.05**	**77.86%**	**0.09**
SVM	**Polarity**	**79.50 %**	**4.34**	**72.95 %**	**12.23**
	Frequency	**80.32%**	**4.22**	**66.39%**	**12.42**

8. Future Work

For future work, we would like to conduct our experiment on a larger more representative dataset.

In addition, sentiment cannot be separated from semantic. Counting words or recognizing the polarity of certain terms without making sense of the semantic may hide lots of information. Some positive terms can be used ironically to express negative ideas and some negative terms can be used informally to express extreme positive emotions. In such scenarios semantic means a lot. Some research efforts were made to incorporate the semantic in training the classifiers by means of using sentiment and semantic topics [22]. In the future we will consider incorporating semantic features in our evaluation of classification algorithms for Twitter sentiment analysis.

9. Conclusion

Twitter is one of the most popular social networks where users can tweet about different topics within the 140-character limit. This small size of text imposes a significant challenge to Twitter sentiment analysis since tweets datasets are often too sparse.

In this paper, we have designed an evaluation method for evaluating the effect of different input representations and formats on the performance of the classifiers. Hence, we provided formal performance evaluation of sentiment classification based on fair experimental setups.

The experimental results show that Multinomial Naïve Bayes classifier outperformed other classifiers examined in the study in the context of Twitter sentiment analysis being less affected by the sparsity of Twitter dataset. Unigrams as a form of representing dataset feature proved to be more effective in the context of Twitter sentiment analysis as they produce less sparse datasets. From our experiments, we could not get proof on best choice for frequency vs. polarity representation of data. Finally, despite the strong capabilities of SVM, it generated the least accuracy results taking the longest processing time, it proved to be negatively affected by data sparsity, making it less preferable choice for Twitter sentiment analysis.

For future work, we would like to expand the scope of our experiments and run the classifiers on more than one dataset considering number of different languages in order to have more representative inputs and thus better generalizable results.

References

1. Sterling, G., "Survey: 91 Percent Have Gone Into Stores Because Of Online Promotion," Marketing Land, December 2012. [Online]. Available: http://marketingland.com/survey-91-percent-have-gone-into-stores-because-of-online-promotion-28796. [Accessed November 2015].
2. DeMers, J., "How Social Media is Changing the SEO Industry," Search Engine Journal, April 2013. [Online]. Available: http://www.searchenginejournal.com/how-social-media-is-changing-the-seo-industry/63060/. [Accessed November 2015].
3. Olensky, S., "Are Brands Wielding More Influence In Social Media Than We Thought?," Forbes, May 2012. [Online]. Available: http://www.forbes.com/sites/marketshare/2012/05/07/are-brands-wielding-more-influence-in-social-media-than-we-thought/. [Accessed November 2015].
4. Pang, B., & Lee, L, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval,* vol. 2, no. 2, p. 1–135, 2008.
5. Henschen, D., "Seven Shades of Sentiment," *Information Week 1337,* 2012.
6. Vinodhini, G., & Chandrasekaran, R. M, "Performance Evaluation of Machine Learning Classifiers in Sentiment Mining," *International Journal of Computer Trends and Technology (IJCTT),* vol. 4, no. 6, 2013.
7. Cui, H., Mittal, V., & Datar, M, "Comparative Experiments on Sentiment Classification for Online Product Reviews," *AAAI* , vol. 6, pp. 1265-1270, 2006.

8. Abbasi, A., Chen, H., & Salem, A., "Sentiment Analysis in Multiple Languages: Features selection for Opinion Classification in Web Forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, pp. 1-34, 2008.
9. Bollen, J., Pepe, A., & Mao, H, "Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena," in *International AAAI Conference on Weblogs and Social Media*, Barcelona, 2009.
10. Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T, "Target-dependent Twitter Sentiment Classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, 2011.
11. Go, A., Bhayani, R., & Huang, L, "Twitter Sentiment Classification using Distant Supervision," Stanford Universty Press, Stanford, 2009.
12. Speriosu, M., Sudan, N., Upadhyay, S., & Baldridge, J, "Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph," in *Proceedings of the First workshop on Unsupervised Learning in NLP*, UK, 2011.
13. Chaovalit, P., & Zhou, L, "Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, Hawaii, 2005.
14. Elhawary, M., & Elfeky, M, "Mining Arabic Business Reviews," in *Data Mining Workshops (ICDMW)*, Sydney, 2010.
15. Wilson, T., Wiebe, J., & Hoffmann, P, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," in *In Proceedings of the conference on human language technology and empirical methods in natural language processing*, Vancouver, 2005.
16. Perez-Tellez, F., Pinto, D., Cardiff, J., & Rosso, P, "On The Difficulty of Clustering Company tweets," in *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, New York, 2010.
17. Bhuta, S., Doshi, A., Doshi, U., & Narvekar, M, "A Review of Techniques for Sentiment Analysis of Twitter Data," in *Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2014.
18. Hongbo, D., Data Mining Techniques and Applications An Introduction, C&C Offset China, 2010.
19. Manning, C. D., Raghavan, P., & Schütze, H, Introduction to information retrieval, Cambridge : Cambridge University Press, 2008, pp. 253-287.
20. Yu, H., & Kim, S., "SVM Tutorial—Classification, Regression and Ranking.," Springer Berlin Heidelberg in Handbook of Natural Computing, 2012.
21. Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, "Thumbs up: Sentiment Classification Using Machine Learning Techniques," Association for Computational Linguistics, 2002.
22. Saif, H., He, Y., & Alani, H, "Alleviating Data Sparsity for Twitter Sentiment Analysis," in *CEUR Workshop Proceedings (CEUR-WS. org)*, Lyon, France., 2012.

Event Detection in Czech Twitter

Václav Rajtmajer[1] and Pavel Král[1,2]

[1] Dept. of Computer Science & Engineering
Faculty of Applied Sciences
University of West Bohemia
Plzeň, Czech Republic
[2] NTIS - New Technologies for the Information Society
Faculty of Applied Sciences
University of West Bohemia
Plzeň, Czech Republic
{rajtmajv,pkral}@kiv.zcu.cz

Abstract. The main goal of this paper is to create a novel experimental system for the Czech News Agency (ČTK) which is able to monitor the current data-flow on Twitter, analyze it and extract relevant events. The detected events are then presented to users in an acceptable form. A novel event detection approach adapted to the Czech Twitter is thus proposed. It uses user-lists to discover potentially interesting tweets which are further clustered into groups based on the content. The final decision is based on thresholding. The main research contribution is to propose an original approach to harvest potential events from Twitter with high download speed. We experimentally show that the proposed approach is useful because it detects a significant amount of the events. It is worth of noting that this approach is domain independent.

Keywords: Clustering, Event Detection, Twitter

1 Introduction

The Czech internet society is growing every year. One way to share information with the others are social networks which are represented in our case by Twitter. We have chosen Twitter because of its large size, significant amount of other existing work about this network and the needs of our client. However, this work could be also used on other social networks.

Twitter uses very short messages (limited by 140 characters) which are posted on-line as status updates, so-called *tweets*. The tweets can be accompanied by photos, videos, geolocation, links to other users (words preceded by the sign @) and trending topics (words preceded by the sign #). The posted tweet can be liked, commented by the other tweets, or redistributed by other users by forwarding, so-called *retweet*. Due to its simplicity and easy access, Twitter contains a very wide range of topics from common every day conversations over sport news to news about an ongoing earthquake.

As already stated, Twitter is an interesting source of on-line information which is often used for further analysis and data-mining. Therefore, it can be also employed for

automatic real-time event detection. This is very useful for many journals and particularly for news agencies in order to be the first to publish new interesting information.

Several definitions of an event exist, however we use for this work the definition from a Cambridge Dictionary. It defines an event as "anything that happens, especially something important and unusual[3]".

The main goal of this paper is thus to propose a novel approach for Czech Twitter analysis in order to discover new events in real-time. The proposed method will be domain independent and adapted to the characteristics of the Czech Twitter. Therefore, it respects the properties of the Czech language and behaviour of the Czech Twitter users. The first issue is relatively easy to handle using natural language processing methods. However, the second one is more difficult, because the activity on the Czech Twitter is significantly lower than on the other languages, which is particularly evident for English or French. Therefore, classical statistical methods can suffer on this task. It is worth of noting that, to the best of our knowledge, no other approach/system for automatic event detection in Czech Twitter exists.

The core of the proposed method consists in using user-lists to download a sufficient number of Czech tweets in real-time. Then, we discover potentially interesting tweets which are further clustered into groups based on the content. The final decision is based on thresholding.

The rest of the paper is organized as follows. Section 2 is a short review of Twitter analysis with a particular focus on event detection methods. Section 3 presents our event detection approach which is adapted to the Czech Twitter. Section 4 deals with the results of our experiments and also with the usability of the system. In the last section, we conclude the experimental results and propose some future research directions.

2 Related Work

Twitter with its popularity offers many possibilities for data processing and analysis, therefore numerous studies have investigated Twitter. For instance, it is used in [1] as a data source of sentiment analysis and opinion mining. The authors have collected a sentiment analysis corpus from Twitter and they have further built an efficient sentiment classifier on this corpus. Another work dealing with sentiment analysis from Twitter is proposed in [2]. This paper investigates the importance of linguistic features for sentiment detection with a good accuracy.

The data in this network can be also investigated for sociological surveys as shown for instance in [3]. The authors have analyzed a group polarization using the data collected from dynamic debates. Another study analyzes Twitter community [4] to discover user activities. A taxonomy characterizing the underlying intentions of the users is presented.

Twitter can be also successfully used for event detection as presented in the following text. Sakaki et al. propose in [5] an algorithm to monitor tweets and detect target events from Twitter. The proposed approach is interesting, however it is domain dependent. It uses some keywords to characterize specific events (e.g. earthquake or

[3] http://dictionary.cambridge.org/dictionary/british/event?q=event

typhoons). The proposed system is very important because it may warn people about incoming disaster.

Earle et al. present in [6] a simple earthquake detector. The proposed approach is based on a rapid increase of the frequency of tweets containing the word "earthquake" or its equivalents in other languages. The two previous approaches are domain dependent and therefore it is not possible to be used for general event detection.

Li et al. deal in [7] also with event detection from Twitter. They propose a sophisticated system called *Twevent*, which first detects "bursty tweet segments" as event segments and then they are clustered considering both their frequency distribution and content similarity to discover events. Wikipedia is used as a knowledge base to derive the most interesting segments to describe the identified events and to identify realistic events. The difference of this system from the two previous ones is that it is domain independent. Therefore, it can discover all types of event.

Petrovic et al. present in [8] an interesting first story detection algorithm from Twitter stream. Their event is meant as a new topic which has never appeared in Twitter before. The proposed method is based on locality-sensitive hashing.

Another work [9] presents a lightweight event detection system which analyzes wavelet signal of hashtag occurrences in the Twitter stream. It also describes the detected events by a Latent Dirichlet Allocation topic inference model based on Gibbs Sampling.

The approach proposed by Weng et al. in [10] uses wavelets for event detection. Wavelets are used to analyze the frequency-based raw signals of the words. Non significant words are then removed by looking at their corresponding signal auto-correlations. The remaining words are then clustered into event clusters with a modularity-based graph partitioning method.

For additional information about the techniques for event detection in twitter, please refer the survey [11].

The event detection methods described above are focused particularly on English. Relatively few works are oriented to other languages. However, to the best of our knowledge, no other work for automatic detection of events in Czech Twitter exists.

3 Event Detection Adapted to the Czech Twitter

The proposed method is composed of three main functional units (*Tweet Stream Analysis*, *Preprocessing* and *Event Detection*) which are further decomposed into six tasks as depicted in Figure 1.

The first task, *data acquisition*, is beneficial to harvest on-line appropriate data from Twitter for a further processing. Then, *spam filtering* is done to remove tweets with useless information (so called "spam"). The third task is *lemmatization* which is used for word normalization. The next step is *non-significant word filtering*. While the previous filtering was at the tweet level, this one is at word level and is used to remove non-significant words which could decrease the detection performance. The next step to discover events is *clustering*. We group together the tweets with similar content using a clustering method. The final decision about an event is based on the thresholding.

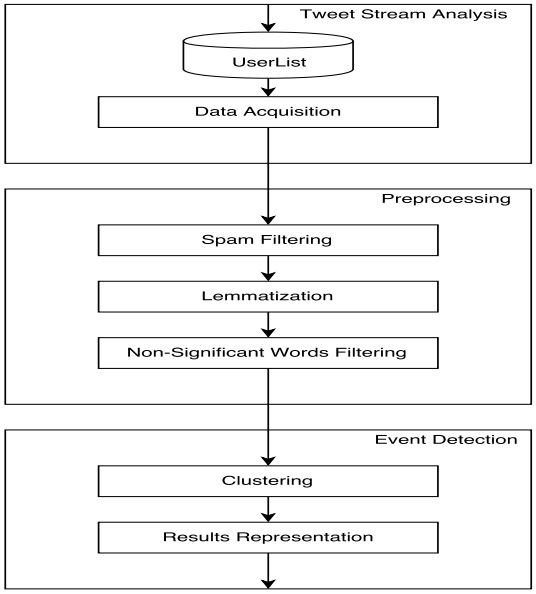


Fig. 1. Overall scheme of the proposed method

The last step, *results representation*, is used to show the detected event to users in an acceptable form. All these steps are in detail described below.

3.1 Twitter Stream Analysis

Data Acquisition We have analyzed different possibilities of the Twitter API to get the maximum possible number of Czech tweets. We must get a significant amount of tweets in Czech languages for free. Unfortunately, it is not possible to obtain only the Czech tweets by language constraints, because of the Czech constraint is missing. There is available only "sk" field which contains Czech and Slovak tweets together.

Therefore, we have decided to filter Czech tweets according to geolocation. As our area of interest we have chosen a square region, covering most of the territory of the Czech Republic. We have analyzed the download rate of the Czech Twitter by this method with evaluation interval from 22 August 2015 to 29 August 2015. Figure 2 shows the results of this analysis compared with the French Twitter. This figure shows that the activity of the French Twitter is more than 10 $\times$ higher than the Czech Twitter. The average of the Czech download rate is 495 tweets/hour. However, after a detailed examination, we have identified that only less than 20% is written in Czech languages. Unfortunately, this number is insufficient for a successful event detection in real-time. Therefore, we have proposed a novel data acquisition method based on *UserLists*.

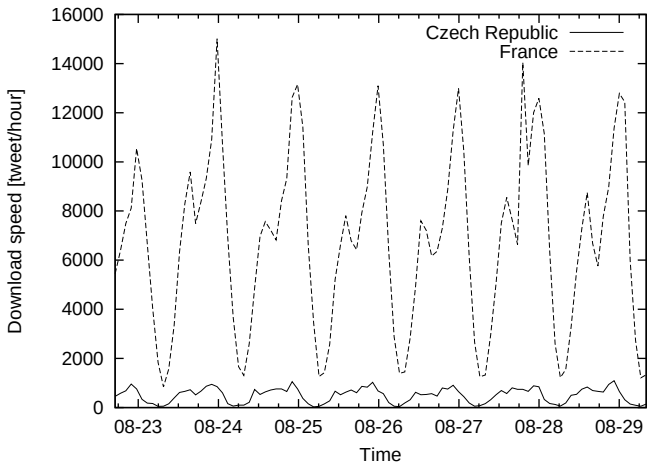


Fig. 2. Czech republic vs. France download speed

UserList We have proposed a method which uses UserLists for acquisition of the significant amount of tweets that contain potential events. This method is motivated by the fact that about 20% of Twitter users are posting informative tweets, whereas the remaining 80% not [12].

UserList is a Twitter possibility to allow each user to create 20 lists with an option to store up to 5,000 users into one list. These lists can be used to show all tweets that these users have posted and this procedure can be used with Twitter API to get all published data from 100,000 particular users.

We have experimentally shown that this method provides several times more data than the typical methods proposed by Twitter (Search, FilteredStream). The results of this experiment are shown in Table 1. This table shows that the proposed method outperforms both other methods more than 6 times.

Table 1. Download speed comparison of the different methods on Czech Twitter

Twitter API function	**Tweets no. / hour**
Search	43.5
FilteredStream	56.6
UserList *(proposed)*	324.3

Our issue is now to select the representative users in order to detect appropriate events. Our system is designed for general event detection. Therefore it must cover all Twitter topics by active authors from all fields. We use a small sample of interesting

people provided by Czech News Agency (ČTK[4]) and this sample is automatically extended by our algorithm.

The algorithm to complete the UserList is based on the assumption that:

- We have already a representative group of the users (sample provided by ČTK);
- this set covers a representative part of our domain of interest;
- their followers would be the users with similar interests.

Therefore, we get by the Twitter API detailed information about all the followers of our initial group. Then, we filter out all foreign (no Czech) users and we continue with the first step. Our algorithm is stopped when a requested number of the users is explored.

For every user u, it is then computed a rank R_u which is based on its number of followers Fn and the number of submitted tweets Tn as follows:

$$R_u = w.Fn + (1 - w).Tn \tag{1}$$

where w is the importance of both criterions and was set experimentally to 0.5.

Our list is sorted by this rank and the "best" 100,000 users are added to our twitter lists for a further processing. Twitter ecosystem is very dynamic and it evolves very quickly. Therefore, this list must be periodically updated to keep actual information.

3.2 Pre-processing

Spam Filtering As already stated, this task is realized in order to remove tweets with useless information. These tweets are filtered with a manually defined set of rules (or with a list of entire tweets). Table 2 shows some examples of whole tweets. The rules are based on the predefined patterns.

Table 2. Examples of tweets to filter

Tweet	English translation
Automatically created messages	
Přidal jsem novou fotku na Facebook.	I have added a new photo on Facebook.
Líbí se mi video @YouTube.	I like @YouTube movie.
Označil(-a) jsem video @YouTube.	I have marked @YouTube movie.
(Everyday) useless tweets created by the users	
Dobré ráno!	Good morning!
Jdu obědvat, dobrou chuť.	I'm going to have lunch, enjoy your meal.

Of course, this simple method does not filter all useless tweets. However, we assume that they will not be detected as events by our detection algorithm due to their not significant amount. Therefore, it is not necessary for the current system to implement more sophisticated filtering algorithm.

[4] http://www.ctk.eu/

Lemmatization Lemmatization consists in replacing a particular (inflected) word form by its lemma (base form). It decreases the number of features of the system and is successfully used in many natural language processing tasks. We assume that lemmatization can improve the detection performance of our method. It can be useful particularly in clustering to group together appropriate words.

Following the definition from the Prague Dependency Treebank (PDT) 2.0 [13] project, we use only the first part of the lemma. This is a unique identifier of the lexical item (e.g. infinitive for a verb), possibly followed by a digit to disambiguate different lemmas with the same base forms. For instance, the Czech word "třeba", having the identical lemma, can signify *necessary* or *for example* depending on the context. This is in the PDT notation differentiated by two lemmas: "třeba-1" and "treba-2". The second part containing additional information about the lemma, such as semantic or derivational information, is not taken into account in this work.

Non-Significant Word Filtering Non-significant words (also sometimes called stop words) are considered words with high frequencies which have in a sentence rather grammatical meaning as for instance prepositions or conjunctions. In this version, the filtering is based on a manually defined list. We plan to implement more sophisticated method based on Part-of-Speech (POS) tags in the further version. However, we assume that this improved removal will play marginal role for event detection.

3.3 Event Detection

Clustering After getting the data we are facing the problem of extracting events. We use a clustering technique for this purpose. Consider that we get in real-time the filtered and lemmatized tweets which can represent due to the UserList method very probably the events. We transform every tweet into a binary representation using a bag of words method, which represents its unique location in n-dimensional space. Then the clustering algorithm is as follows:

1. take an (unprocessed) tweet
2. calculate the cosine distance between a vector representing this tweet and all the others
3. choose a closest tweet (or cluster of tweets if any) and group them together (the maximal allowed distance is given by the *threshold* Th)
4. repeat the two previous operations (*go to step 1*) till all tweets are processed

The clusters created by this algorithms represent the events. Of course, the clustering does not guarantee that the created clusters represent only the events. This is done by the pre-processing:

- UseList data acquisition method harvests particularly informative tweets which contains mainly the events;
- Spam filtering step removes several useless tweets (no events).

We also define a parameter T, which indicates a time period for the clustering. We assume that different events will be produced at different "speed" (different activities of Twitter users). For instance, information about the winner of the football championship can be quicker (more contributions in a short period) than information about a new director of some company.

It is worth of noting, that we have also considered a *gradient* of the frequencies in some event clusters. Unfortunately, this improvement did not work because of the small activity of the users on the Czech Twitter.

Results Representation The results of the clustering are thus the groups of tweets with some common words. This group is represented by the *most significant* tweet. This tweet is defined as a message with the maximum of common words and the minimum of the other words. This representation is used due to the effort to use an answer in natural language, instead of a list of key-words or a phrase.

4 Experimental Results

This section describes the experiments realized for validation of the proposed event detection method. This evaluation was done off-line. Therefore, we have saved 15,856 tweets from two day period using the novel proposed method based on UserLists (see Sec. 3.1). This approach was used because, as already proved, this algorithm provides about 6 times more relevant data than the other methods.

Then, we have executed our system with different values of the acceptance threshold ($Th \in [0; 1]$) and analyzed the results.

We have analyzed the resulting clusters obtained by our system. This analysis have shown that for results with $Th > 0.5$ the system still detects the majority of events correctly (high *precision*). However, the main interest is to have the *recall* as high as possible and the precision is not so important, because of the possibility of manual filtering of incorrectly detected events. Therefore, we set in our system a lower acceptance threshold which causes to detect more events with some false positives.

These preliminary results were shown and discussed with our client who is ready to test this experimental version of the system. It is clear that the current version will already help to the reporters to reduce their work with manual checking of the available data sources.

One sample of the results is depicted in Figure 3. This figure shows that six tweets are saved by our acquisition method (right). They are then clustered into two groups containing three and two tweets (left "bubbles"). Finally, one representative tweet is chosen from both clusters to be presented to the user (bold text left).

5 Conclusions and Perspectives

The main goal of this paper was to create an experimental system for ČTK which is able to monitor the current data-flow on Twitter, analyze it and extract relevant events. We have thus proposed a novel domain independent event detection approach adapted

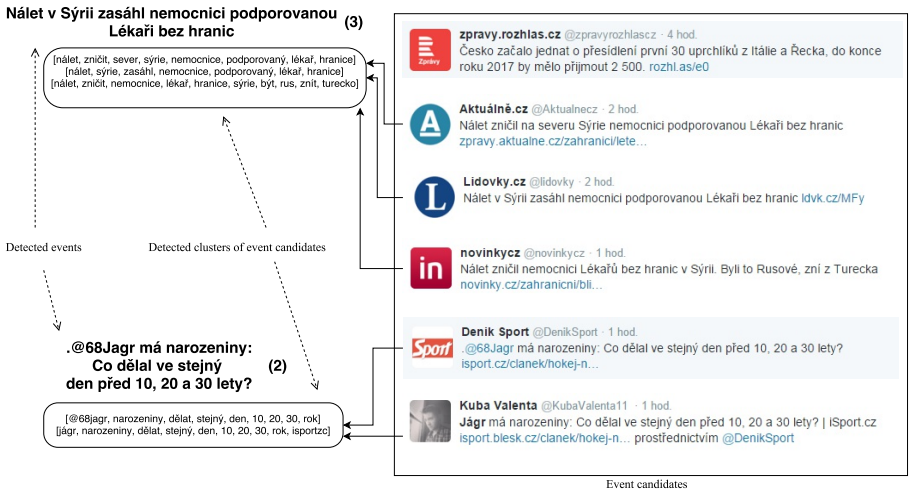


Fig. 3. Event detection example (time period $T = 2h$ and acceptance threshold $Th = 0.5$)

to the Czech Twitter. The main research contribution consists in proposing an original approach to harvest potential events with high download speed. This method uses user-lists to discover potentially interesting tweets which are further clustered into groups based on the content. The final decision is based on thresholding. We have experimentally shown that the results are very promising because we detect a significant amount of potential events.

The first perspective consists in improving our clustering method using more sophisticated semantic similarity functions. Another perspective is adaptation and evaluation of the proposed system on other European languages.

Acknowledgements

This work has been partly supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports and by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications.

References

1. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREc. Volume 10. (2010) 1320–1326
2. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: The good the bad and the omg! Icwsm **11** (2011) 538–541
3. Yardi, S., Boyd, D.: Dynamic debates: An analysis of group polarization over time on Twitter. Bulletin of Science, Technology & Society **30** (2010) 316–327
4. Java, A., Song, X., Finin, T., Tseng, B.: Why we Twitter: An analysis of a microblogging community. In: Advances in Web Mining and Web Usage Analysis. Springer (2009) 118–138

5. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, ACM (2010) 851–860
6. Earle, P.S., Bowden, D.C., Guy, M.: Twitter earthquake detection: earthquake monitoring in a social world. Annals of Geophysics **54** (2012)
7. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: Proceedings of the 21st ACM international conference on Information and knowledge management, ACM (2012) 155–164
8. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to Twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 181–189
9. Cordeiro, M.: Twitter event detection: Combining wavelet analysis and topic inference summarization. In: Doctoral Symposium on Informatics Engineering, DSIE. Volume 56. (2012)
10. Weng, J., Lee, B.S.: Event detection in Twitter. ICWSM **11** (2011) 401–408
11. Atefeh, F., Khreich, W.: A survey of techniques for event detection in Twitter. Computational Intelligence **31** (2015) 132–164
12. Naaman, M., Boase, J., Lai, C.H.: Is it really about me?: message content in social awareness streams. In: Proceedings of the 2010 ACM conference on Computer supported cooperative work, ACM (2010) 189–192
13. Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtskỳ, Z., Hajič, J.: Hamledt: Harmonized multi-language dependency treebank. Language Resources and Evaluation **48** (2014) 601–637

Blog Annotation: From Corpus Analysis to Automatic Tag Suggestion

Ivan Garrido-Marquez*, Jorge Garcia Flores,
François Lévy, and Adeline Nazarenko

LIPN, Paris 13 University – Sorbonne Paris Cité & CNRS, France,
99, av. J.-B. Clément, F-93430, France
{garridomarquez,jgflores,fl,
adeline.nazarenko}@lipn.univ-paris13.fr

Abstract. Nowadays, blogs cover a large audience and they become part of mainstream media. Tags and categories are structural elements of a blog post intended to increase a blog's visibility and enhance navigation and searching. We suppose that those annotations are made on subjective grounds rather than in a systematic way. This paper presents a 11 million words corpus of blogs posts in French dedicated to the analysis of blog post tagging and categorization practices. We present experiences on automatic tag and category suggestion based on this corpus. Preliminary results show that around 27% of the overall tags can be predicted from lexical frequency analysis of blog posts. Furthermore, a first comparison experience with an existing tag suggestion tool shows that an important proportion of the tags used for blog description are not present in the blog post. Preliminary observations of annotations in time might suggest that taking into account diachronic information from previous post might improve the tag suggestion process.

Keywords: Annotation, blogs, tag suggestion, tagging, corpus of blogs, corpus analysis

1 Introduction

The rise of blogs followed that of the web at the end of the 90's. The blogosphere boomed in the early 2000s and it has been part of the mainstream media for more than a decade. Originally a blog was mostly a personal journal published on a website containing multiple entries. It is estimated that 152 millions blogs were active by the end of 2010 [2]. In the single Wordpress domain, 22.8 billion pages are viewed every month while 56 millions of new posts are published. Nowadays, the blogosphere and its hundreds of millions of blogs has become an essential mean for sharing information.

* This work has been partially funded by the French Ministry of National Education, Higher Education, Training and Scientific Research and is supported by the French National Research Agency (ANR-10-LABX-0083) in the context of the Labex EFL.

The elemental unit of blogs is the post, a piece of content normally written by one single author. To ease the classification of blog posts, it is very common that bloggers annotate them with categories and/or tags. In the first case, the blog classify its contents into a predefined set of categories, each category corresponding to a group of posts that are somehow related. Category systems may be organized into taxonomies where posts can be classed in more than one category. A different way to mark a post is by adding keywords, called tags, that somehow summarize the topic. These tags come from an open vocabulary, and both their variety and their amount might grow as the blog post number increases. The evolution of the blog's subject might enrich the variety of these tags as well.

Annotating blog posts with tags makes easier the searching of content in the blog. It enhances navigation as well, by allowing to group posts of a particular subject or related content. They might increase blog's visibility in the web by letting web search engines index them with tags. On the other hand, adjoining tags or categories is mostly based on a distributed, subjective or any other arbitrary criteria. We suppose that the study of tagging and categorization practices could bring a better understanding of the semantic relations underlying between tags, categories and posts.

The paper is structured as following: In section 2 we present related work on blog post analysis and tag and category prediction. Section 3 presents an 11 million words corpus of blog posts in French on law, cooking and technology. Section 4 introduces our first experiences on tag and category prediction. Section 5 discusses the results while section 6 gives a hint of future perspectives for this work.

2 Previous works

Even if "one of tagging's biggest appeals is its simplicity and ease of use" [1], we tend to believe that the resulting annotations are not systematic at all because they almost always depend on the user, as the example 1 from section 4.2 suggests.

We distinguish three types of approaches proposed to automatically identify possible tags for a blog post:

1. Predicting tags from a fixed set by using machine learning. In [5] Katakis and al. present a system for tag recommendation in social bookmarks. The system recommendations are meant to be particular for each author: it recommends the most popular tags present in the post and previously associated to the user.
2. Computing a topic description over the set of tags [10,6,4,7]. For example, in [10] Tsai uses topic modeling for mining the tags in blogs according to topics. Each tag is represented by a probability distribution over topics, and each topic represented by a probability distribution over terms for that topic. The technique is based on LDA for topic modeling and dimensionality reduction.

The most suitable terms for tagging can be identified by computing the topics.

3. Searching for tags on similar posts. The system AutoTag described in [8] estimates similarity between blog posts with information retrieval measures and selects the most similar posts to the one at hand. Then it extract a list of tags ranked by their frequency in the selection of posts. At the end, a filtering and reranking step boosts the score of tags previously used by the user, and then the best tags are proposed.

There are various developed tag suggestion tools, available as APIs or web services, to help bloggers to annotate their posts. They rely on post content and propose the most relevant extracted keywords as tags. *Zemanta*[1], *Yahoo! Content Analysis*[2] and *Open Calais*[3] can match the tags with entities in external descriptive resources. Others like *AlchemyAPI*[4] and *Thoth*[5] can make use of blog-level statistics or sophisticated natural language processing techniques. They are designed as independent tools, but provide plugins for major blog platforms. Some like *Climate tagger*[6] work for specific domain content such as documents abour climate.

Despite the existing annotation methods and tools, blog annotation often remains manual and unsystematic, which might hinder the usability of the blogs. With 10 years of hindsight, we can analyze blog annotation in the long term so as to understand how tag suggestion works and why the use of tag annotation tools isn't as widespread practice as one could expect from the huge amount of bloggers. This historical perspective, impossible ten years ago, allows a long term analysis, which we consider the most important value of the corpus.

3 Corpus analysis

Our corpus includes around 11 millions of words. It is composed of 20 blogs in French dealing with different topics. In order to analyze the annotation practices, we focus on blogs containing tag and category annotations. We also focus of textual parts only, leaving aside the images and video contents. The main topics included in the blogs are cooking, law, technology, and video games. From Table 1 we can observe high standard deviations on every description feature: number of authors, categories, tags and size. Some blogs have a lot of contributing authors, up to 143, while many of them have a single author (9) or less than 5 authors (13). The number of the posts as well as their size also vary significantly (from 184 to 6,585 posts per blog and from 55 to almost 1,300 words per post). It is worthy to notice that four of these blogs contain more than 1 million of words each.

[1] http://www.zemanta.com/
[2] https://developer.yahoo.com/contentanalysis/
[3] http://www.opencalais.com/
[4] http://www.alchemyapi.com/
[5] https://fr.wordpress.org/plugins/thoth-suggested-tags/
[6] http://www.climatetagger.net/

Table 1. Corpus description

Blog	posts	authors	cats	tags	Kb	words
jeuxvideo6	184	6	18	556	968	66,991
technologie2	243	1	38	40	1108	55,073
droit3	283	1	13	77	3704	366,816
technologie5	305	1	16	295	2112	177,034
technologie3	343	13	41	397	2120	193,160
technologie5	374	2	25	358	2816	317,551
cuisine3	474	1	50	243	2048	152,377
droit1	485	2	4	84	4736	466,702
cuisine1	514	1	60	460	2180	133,063
technologie4	573	1	12	321	2508	110,111
jeuxvideo5	1135	2	37	2467	5716	387,632
cuisine2	1166	1	26	695	10064	1,051,706
jeuxvideo1	1423	3	43	1772	9672	868,019
technologie1	1423	17	56	1231	6740	416,498
jeuxvideo4	1501	17	40	3146	9048	698,151
cuisine4	1721	1	25	265	9092	891,033
droit4	1752	1	15	0	14104	1,333,494
droit2	1769	143	48	741	10440	771,041
jeuxvideo2	2483	6	33	2978	17060	1,349,318
jeuxvideo3	6587	67	91	4650	31148	1,598,143
average	1236.9	14.35	34.55	1038.8	7369.2	570,195.65
std dev	1426.12	33.77	20.57	1292.08	7221.87	475,284.97
max	6587	143	91	4650	31148	1,598,143
min	184	1	4	0	968	55,073
total	24738	287	691	20776	147384	**11,403,913**

3.1 Annotation activity

Table 1 shows stats about the annotation activity. The average number of categories is 34 with a high standard deviation. Categories for a single blog range from 4 to 91. Tags for a single blog range from 40 to 4650. The standard deviation for tags is high as well. Table 2 reviews the annotation activity at a *per blog post* level. The mean number of categories per post ranges from 1 to 4.27.

Overall, each blog has its own annotation profile. Furthermore, the tagging activity might be more arbitrary than the category attribution, so one could wonder if a more consistent semantic annotation system is possible for the blog annotation activity by using a tag suggestion tools, such as those cited in Section 2. On the other han, figures suggest that categories are semantically more structured than tags.

We suppose that the tags arise from a wide variety of sources: the post content, the pool of existing tags, external resources (the web, another blog, search engines): *i.e.* new tags do not seem derived from the post content.

Table 2. Categories and tags per post

	Categories				Tags			
Blog	**mean**	**min**	**max**	σ	**mean**	**min**	**max**	σ
cuisine1	1	1	1	0	2.12	0	17	3.42
cuisine2	1	1	1	0	5.45	1	20	3.51
jeuxvideo1	3.41	1	11	1.87	4.95	0	19	1.69
technologie1	1.07	1	6	0.29	3.16	0	16	3.55
technologie5	2.31	1	8	1.22	4.20	0	24	4.34
droit1	1	1	1	0	2.41	0	6	1.31
jeuxvideo2	4.27	1	12	1.74	8.84	1	21	3.19
technologie2	1.88	1	5	0.96	0.79	0	5	1.09
jeuxvideo3	0.99	0	1	0.07	4.09	0	28	2.24
jeuxvideo4	2.22	1	3	0.71	6.07	0	45	3.92
droit2	1.72	1	31	2.81	3.19	0	19	2.82
cuisine3	1	1	1	0	5.20	1	14	1.88
technologie3	1.31	1	4	0.60	2.54	0	6	1.34
droit3	1.41	1	5	0.68	2.94	0	9	2.28
cuisine4	1	1	1	0	4.04	0	11	1.68
droit4	3.14	1	7	1.08	0	0	0	0
technologie4	1	1	1	0	3.13	0	13	2.1
jeuxvideo5	2.94	1	10	1.22	3.79	0	13	1.7
technologie5	4.18	1	12	2.03	6.72	0	18	3.17
jeuxvideo6	1.01	1	2	0.1	5.34	0	20	2.84

3.2 Evolution over time

Our corpus covers a period of 10 years of blogging, from 2005 to 2010. It allows to study the temporal evolution of the blog annotation practices. Let's consider the example of the *droit2*, which was launched in 2007 and is still active in 2016.

Figure 1 shows the distribution of the posts over 10 years (120 months) of the *droit2*. A peak of the blog post activity can be observed between Months 37 and 75. This blog has the highest number of authors (see Table 1), but an average number of categories (1.72) and tags (3.19) per post. Between 2008 and 2011, posting activity was very intense and by a high number of authors, which didn't imply higher than average tagging nor categorizing variety. The 143 authors of the *droit2* annotated with less tags and less categories per post than the single author of the significantly smaller *technologie5* blog.

Figure 2 shows three different tag profiles for blog *droit2*. Like most of the categories, they present irregular distributions. One could argue that it is related to the news and that *Contrefaçon*, for instance, appears and disappears in the annotations because of an underlying issue that becomes hot and then fades in the news. However, we observe that the histogram of *Contrefaçon* actually follows the activity in the blog (measured as the number of posts per month, on Figure 1). It therefore reflects a rather stable distribution. On the opposite, the histogram of *Avocats* shows a surprinsingly uneven distribution for a term like

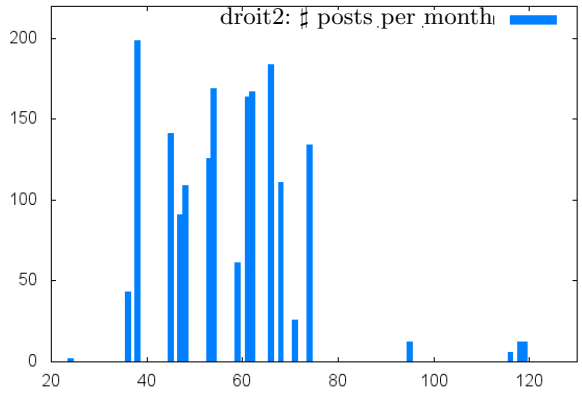


Fig. 1. Distribution of the *droit2* posts over a period of 100 months. X-axis is the date in months (from the 20^{th} to the 120^{th} month of the global blog corpus). Y-axis is the number of posts.

lawyers, which would be expected to be very common in a legal blog. And also, *Internet* has the same profile as *Avocats* at the same time, hinting to a possible correlation.

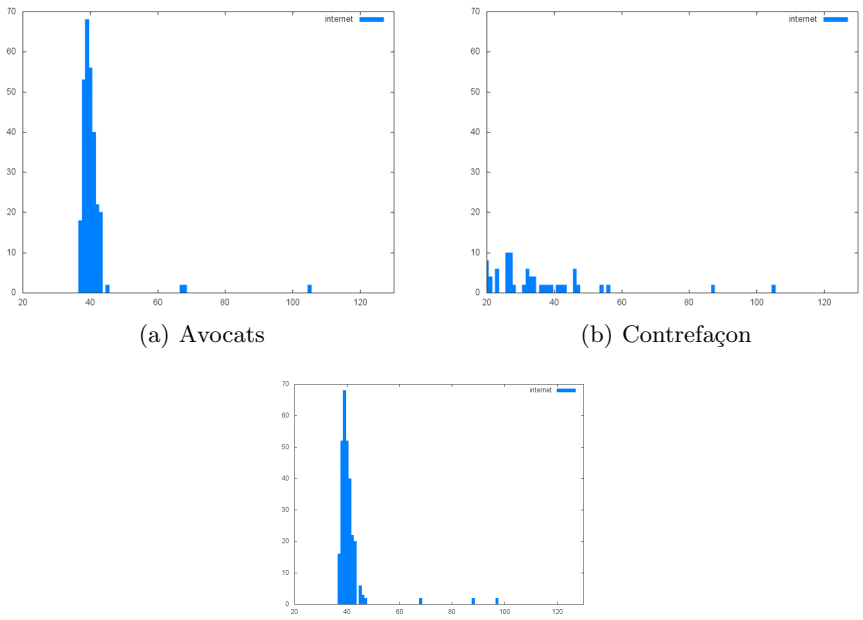
These observations suggests that the tagging and categorizing activity are not indexed to the post frequency over time. From these figures we suppose that the blog annotation activity is rather arbitrary and not very systematic, which limits the utility of tags and categories while searching information withing blogs. It is therefore important for bloggers to be assisted with tag and category suggestion tools.

4 Annotation strategies for blogs

4.1 Tagging from post contents: term frequency strategy

Tag prediction based on word frequency is a traditional approach for tag suggestion [1].We analyse a prediction strategy based on simple term frequency; *tf-idf*; and the combination of both, giving a higher weight to the tags present in the first two strategies. Ten tags were automatically generated and compared with author's hand made tags. Because of the variation on the tags per post, we consider the recall measure (R@10) as the most appropriate for this kind of evaluation.

Table 3 shows the results for frequency based strategies. Data in bold show the best and the worst precision and recall measures. All of them got their highest recall on *technologie3* blog, while the worst recall was for *droit1* blog. We interpret a high recall for a blog as a systematic use of the tags that are frequent in the content of blog posts. On the other hand, a low recall is a sign of outside the post lexical choices as tags. Both blogs show a systematic tagging



(a) Avocats (b) Contrefaçon

(c) internet

Fig. 2. Profile of three tags over time on the *droit2*

approach: *technologie3* blog (whose main subject is technology, therefore physical objects) systematically uses tags from the blog posts, while in the *droit1* blog, authors systematically tag with words that do not occur (or with a very low frequency) in the blog posts. Our best strategy was able to find an average of 27% of author's tags.

4.2 Human annotations versus automatic tagging in blogs

As mentioned in Section 2.1, different platforms and services provide multipurpose meta-data extracted from unstructured text content. Specific tools have been developped for automatically suggesting tags to blog authors. In the case of IBM's Alchemy there is a wordpress plugin called AlchemyTagger, which suggests a set of tags based on a JSON or XML response of AlchemyAPI.

We annotated blog posts with tags suggested by AlchemyAPI, selected the top 10 suggestions according to the relevance score provided by this API and compared to author's tags. We will explain some of our observations with the following examples.

The first exampleis a brief article about the legal authorities involved in the creation of a made-in-europe mark for products. It was tagged only with one tag by its author (*"Avocats; John Doe"*). This tag is composed of two terms *Avocats (Advocates/Lawyers)* and the name of a person *John Doe*. The first term is very

Table 3. Term frequency based tagging strategy

	TF			TFIDF			Mix		
Blog	**P@10**	**R@10**	**F@10**	**P@10**	**R@10**	**F@10**	**P@10**	**R@10**	**F@10**
cuisine1	0.14	0.25	0.18	**0.18**	0.32	0.23	0.13	0.35	0.19
cuisine2	0.14	0.28	0.19	0.14	0.28	0.19	0.11	0.32	0.17
jeuxvideo1	0.10	0.21	0.14	0.10	0.21	0.14	0.08	0.26	0.13
technologie1	**0.17**	0.29	0.21	0.16	0.27	0.20	**0.13**	0.31	0.19
technologie5	0.11	0.24	0.15	0.12	0.28	0.17	0.09	0.30	0.14
droit1	**0.02**	**0.09**	**0.04**	**0.02**	**0.10**	**0.04**	**0.02**	**0.11**	**0.03**
jeuxvideo2	0.13	0.16	0.15	0.13	0.15	0.14	0.11	0.20	0.14
technologie2	0.07	0.43	0.11	0.06	0.37	0.10	0.06	0.45	0.10
jeuxvideo3	0.07	0.18	0.10	0.07	0.17	0.09	0.06	0.2	0.09
jeuxvideo4	0.10	0.20	0.13	0.10	0.19	0.13	0.08	0.23	0.12
droit2	0.07	0.18	0.10	0.07	0.15	0.09	0.06	0.20	0.09
cuisine3	0.15	0.29	0.20	0.15	0.29	0.20	0.12	0.33	0.17
technologie3	0.14	**0.54**	**0.22**	0.15	**0.59**	**0.24**	0.11	**0.62**	**0.19**
droit3	0.04	0.10	0.05	0.04	0.10	0.05	0.03	0.12	0.05
cuisine4	0.11	0.28	0.16	0.13	0.33	0.18	0.09	0.34	0.14
technologie4	0.06	0.15	0.08	0.04	0.11	0.06	0.04	0.16	0.07
jeuxvideo5	0.09	0.24	0.13	0.11	0.30	0.16	0.08	0.31	0.13
technologie5	0.11	0.17	0.13	0.10	0.15	0.12	0.09	0.19	0.12
jeuxvideo6	0.09	0.16	0.11	0.10	0.19	0.13	0.08	0.20	0.11
average	0.10	0.23	0.13	0.1	0.24	0.14	0.08	**0.27***	0.12
max	0.17	0.54	0.22	0.18	0.59	0.24	0.13	**0.62**	0.19
min	0.02	0.09	0.04	0.02	0.10	0.04	0.02	**0.11**	0.03
std dev	0.04	0.11	0.05	0.05	**0.12**	0.06	0.03	0.12	0.05

generic. It is related to the general topic of the blog, which is law, and not to the specific topic of the post. The second term, *John Doe* does not correspond to the author, neither to a person mentioned in the text: it belongs to one of the main owners of the blog. We would like to remark that none of these terms in the tag come from the content of the post.

The 10 tags top-ranked by AlchemyAPI for this post are: *"made in", "made in europe", "marquage d'origine", "Direction Générale", "made in france", "régime européen uniforme", "Europe", "position officielle", "Union française", "Après l'UFIH"*[7]. This proposed tag set proposed was extracted from the actual content of the post.

With 4 tags, the second post example, about the payment in advance of legal proceedings in france by buying revenue stamps online, is richer in author's tags. These tags are *"Action en justice", "John Doe", "Doe avocats", "Procédure"*[8]. Only one tag out of the four comes from the content of the post

[7] Translation:*"Made in", "Made in Europe", "origin marking", "General Management", "Made in France", "uniform European system," "Europe," "official position", "French Union", "after UFIH"*.

[8] "Legal Action", "John Doe", "Doe lawyers", "Proceeding".

itself: "Procédure". "John Doe", "Doe avocats" might actually be intended to improve the visibility of the author and his firm by appearing as meta-data that could be indexed by internet searching engines. The set of tags proposed by Alchemy included *"paiement d'un timbre", "Le paiement s'effectue", "Une fois l'achat", "adresse mail", "également possible", "carte bancaire"*[9]. In this case, it is not possible to say that they summarize the topic of the post, but they are elements related to specific parts of the content.

In these examples, the authors of the blog *droit2* don't use tags taken from the post content. Table 3 confirms this observation: the blog *droit2* has a very low recall of tags coming from the post text. The results for both the frequency strategy and the much more complex analysis tool were particularly low, not that they suggest bad quality tags, but because they propose tags extracted from the text of the post, a policy that contrasts with the author tagging. We also observed that for a certain period of two years almost every post in this blog was tagged with the name "John Doe"; this seems to be a policy to increase the visibility of a certain lawyer's firm. Overall, if we consider the direct match with users' tags, the precision and recall of Alchemy suggestions are close to 0. An in-depth evaluation would be needed to on the one hand to evaluate the quality of author tags and on the other hand the existing tag suggestion tools.

4.3 Representing categories: Words or keyword tags

Unlike tags, categories are an established set, well defined before a post is written. Categories can be created at the moment, but in general they come from a closed vocabulary that classing the posts by topics. Categories frequently define taxonomies which can be seen as a very primitive semantic model for the blog. When a blog has a significant amount of examples to represent well enough its current categories, it would be possible to train a supervised classifier to predict the category of a new post.

We trained four popular supervised classifiers relying on vector machines (SVM) with a linear kernel, Multinomial naive bayes (NB), K nearest neighbor with K=5 (5NN), and Random Forest using 25 weak decision tree classifiers (RF). We measured accuracy with a 10-fold cross validation for every blog in the corpus.

Posts were represented as a bag of words with two different sets of features. The first space of features of the training set contains all the post's words without stop words. The second space of training set features contains the all the post's tags. Table 4 presents the results per blog of this experiment for all the classifiers and both feature sets.

The tags used as features seem to have a similar or even better categorization power than the blog post words. This makes sense because at the end tags are supposed to be a good feature set defined by the author to summarize the content of a post, while categories are meant to focus on the topic. Therefore we might

[9] "Payment of a stamp", "Payment done", "Once the purchase", "email address", "also possible", "credit card"

think that categories hold certain semantic relation with tags, so that categories can be represented by tags. It is worth mentioning the blog *droit4* was excluded from this experience because its posts.

Table 4. Supervised learning for post categorization based on words and tags

	Words				Tags			
blog	**SVM**	**NB**	**5NN**	**RF**	**SVM**	**NB**	**5NN**	**RF**
cuisine1	0.71	0.29	0.60	0.61	0.50	0.55	0.17	0.50
cuisine2	0.73	0.27	0.62	0.67	0.79	0.72	0.62	0.80
jeuxvideo1	0.64	0.57	0.64	0.64	0.75	0.75	0.73	0.78
technologie1	0.61	0.35	0.59	0.49	0.73	0.69	0.64	0.72
technologie5	0.96	0.96	0.94	0.96	0.96	0.94	0.96	0.98
droit1	0.96	0.63	0.76	0.95	0.79	0.75	0.76	0.85
jeuxvideo2	0.94	0.54	0.64	0.92	0.93	0.83	0.59	0.92
technologie2	0.52	0.28	0.55	0.44	0.67	0.63	0.63	0.62
jeuxvideo3	0.52	0.21	0.31	0.51	0.58	0.45	0.42	0.63
jeuxvideo4	0.65	0.46	0.59	0.60	0.78	0.72	0.67	0.74
droit2	0.88	0.51	0.55	0.90	0.81	0.59	0.54	0.91
cuisine3	0.52	0.29	0.54	0.54	0.60	0.44	0.42	0.64
technologie3	0.35	0.23	0.49	0.29	0.46	0.36	0.34	0.39
droit3	0.45	0.45	0.59	0.45	0.57	0.48	0.38	0.67
cuisine4	0.83	0.46	0.72	0.59	0.76	0.73	0.64	0.76
droit4	0.81	0.77	0.79	0.77				
technologie4	0.56	0.38	0.23	0.54	0.76	0.72	0.70	0.76
jeuxvideo5	0.58	0.37	0.49	0.53	0.32	0.31	0.17	0.30
technologie5	0.59	0.59	0.63	0.60	0.64	0.63	0.44	0.64
jeuxvideo6	0.42	0.16	0.63	0.26	0.55	0.55	0.39	0.55
average	0.66	0.44	0.59	0.61	0.68	0.62	0.54	**0.69**

We would like to remark that current systems for annotating blogs do not propose categories from the predefined taxonomy of categories the blog holds. We hypothesize this is mostly due to their not taking historical information of the blog into account to do so. only the content of the post to be tagged.

5 Discussion

We consider that a corpus made of ten years of blogging has unique features for the study of annotation practices. First, posts have a chronological order and a significant size. Second, blogs have a collective authorship, which in practical terms means a group of persons trying to tag and classify (that is, to apply an implicit semantic type system) to text fragments. And third, because 10 years of blogging can produce a reasonable amount of data to study the evolution of this relation between a text fragment and a keyword (the tag) or an implicit taxonomy (the category). Last but not least, to our knowledge there's no other

corpus of French blog post with the size nor the topic variety like the one we are presenting in this paper. This two-level structure (tags, somehow arbitrary, and categories, considered as more stable from a semantic point of view) provides a reference to evaluate automatic annotation. In addition to that, it allows the experimental evaluation of hand annotation quality. The comparison of Alchemy's with author's tags shows an empty intersection but, from a subjective perspective, the annotation of Alchemy looks better than that of authors. The TFIDF based tag suggestion system suggests that the posts word frequency plays a role on this implicit and somehow arbitrary semantic type system that authors might have in mind when tagging and categorizing blog posts. In at least one blog from our blog set (*technologie3*), the authors have a consistent policy of tagging with words from the title or the blog posts. In another blog (*droit1*), the authors have adopted the opposite policy: they tag with terms that do not occur in the blog posts. Further experiences on tag quality might give a better insight on the quality of authors' tag annotations and categories. Furthermore, the chronological order of posts and tags might lead to interesting lexical analysis. For instance, how is it that *lawyers* and *internet* have exactly similar distributions over time? Have they been supported by a common topic? Once in use, even out of order tags are not changed and they generally retain at least some low level activity, obscuring the landscape with non informative noise. At the same time, categories seem to evolve slowly, but they need to be divided when their growth is too fast. Furthermore, some theme can be appreciated differently some time after the annotation has been made. For instance, considerations evoking that Great Britain might leave Europe written in 2005 are still interesting now, but may not have been adequately annotated at this time for 2016 readers. Backward annotation might be an important feature to propose to bloggers.

6 Conclusion and Perspectives

In this paper we presented an 11 million word corpus of French blogs which provides a valuable testbed to analyze blog annotation practices. Blogs subject's include cooking, technology, law and video games. The blog corpus gathers ten years of posts with authors, tags and categories in a chronological order. Future experimental perspectives for this resource will focus on the semantic analysis of bloggers annotation practices and its evolution over time. The chronological study of the annotation practice suggests that considering the life span of tags and categories enriches their relations and opens the perspective of a re-annotation process. We also presented a first experiment on tag suggestion for blog posts. It is based on term frequency, the method chosen for the first experiments in English language ten years ago. Our motivation is that a corpus gathering ten years of authors tags and categories on a chronologicaly ordered text covering a wide variety of subjects is a unique experimental platform for the study of annotation practices. The comparison of the actual tags with those proposed by the analyzed strategies in section 4.1 show us that, in order to offer a better annotation system, we should not only rely on the content of the posts.

Based on section 3, we think that the historical archive of posts is also a good source of knowledge to be considered and analyzed to improve the prediction of tags. External knowledge resources could also be exploited to enrich this systems. In future works, we plan to implement a systematic and semantically consistent method for tagging and categorizing blog posts. This would imply an evaluation on author's tags and category quality to answer the underlying question of our work: Are tags arbitrary or systematic? Can authors produce solid categories and tags for their posts in an objective way? Our results suggest that in 27% of the cases a very simple TF-IDF tag suggestion system can extract author's tags from frequent terms contained in the post. We plan to improve this prediction measures and to exploit external semantic sources in order to propose appropriate and consistent tags for blog posts.

References

1. Brooks, C.H., Montanez, N.: Improved Annotation of the Blogopshere via Autotagging and Hierarchical Clustering. Proceedings of the 15th international conference on World Wide Web (WWW 06) pp. 625–632 (2006)
2. Chapman, C.: A brief history of blogging. http://www.webdesignerdepot.com/2011/03/a-brief-history-of-blogging/ (2011), [Marketing, Web Design, WordPress Mar 14, 2011]
3. Chen, Y., Tsai, F.S., Chan, K.L.: Machine learning techniques for business blog search and mining. Expert Systems with Applications 35(3), 581–590 (2008)
4. Christidis, K., Mentzas, G., Apostolou, D.: Using latent topics to enhance search and recommendation in Enterprise Social Software. Expert Systems with Applications 39(10), 9297–9307 (2012)
5. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel Text Classification for Automated Tag Suggestion. Proceedings of the ECMLPKDD 2008 Discovery Challenge (2008) 9(3), 1–9 (2008)
6. Li, F., He, T., Tu, X., Hu, X.: Incorporating word correlation into tag-topic model for semantic knowledge acquisition. 21st ACM International Conference on Information and Knowledge Management, CIKM 2012 pp. 1622–1626 (2012)
7. Li, Z., Xu, C.: Tag-based top-N recommendation using a pairwise topic model. Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration, IEEE IRI 2013 pp. 30–37 (2013)
8. Mishne, G.: AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts. Proceedings of the 15th international conference on World Wide Web (WWW 06) (2006)
9. Sood, S.C., Owsley, S.H., Hammond, K.J., Birnbaum, L.: TagAssist: Automatic Tag Suggestion for Blog Posts. Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM 2007) (2007)
10. Tsai, F.S.: A tag-topic model for blog mining. Expert Systems with Applications 38(5), 5330–5335 (2011), http://dx.doi.org/10.1016/j.eswa.2010.10.025

A Lookup-Free Approach to Knowledge Extraction from News Feeds

James Little[1] and Chris Painter[2]

[1] Department of Mathematics, Çankaya University, Ankara, Turkey
james@cankaya.edu.tr
[2] Meme-Machines.com, Gloucester, England, U.K
Chris.Painter@zigzag.co.uk

Abstract. Identifying topics without also introducing external assumptions is a major challenge for supervised learning techniques, which by definition classify texts according to precepts. Such approaches identify the presence of pre-classified ideas, but cannot identify new ideas. In this paper, we present the results of applying a well understood unsupervised learning technique, in an innovative way, to news feeds analysis. We identify frequent sets of words using the A-Priori algorithm, and grade those sets according to the significance of the Association Rules that they imply. Such sets of words identify common themes in news feeds autonomously, with stopwords as the only added input. We present in detail this methodology and validate it by examining the identified ideas for their ability to identify actual news.

1 Introduction and Related Work

While the Internet has accelerated the means of production and distribution of Information, it has not in equal measure packaged that Information into formats that humans find easy to use. This disparity between our capacity to produce and our capacity to process is not new; Denis Diderot and d'Alembert in the "Encyclopédie" [1] identified it, and Alvin Toffler [2] introduced it to us as "information overload" in his 1970s best seller "Future Shock". The idea had surfaced in management speak in the 1960s but the Internet has brought the size and perpetuity of the problem into ever sharper relief [3].

The problem has further been exacerbated by the continuing philosophical debate as to what constitutes Meaning. Much of the debate in the 20th century had assumed that meaning was carried in grammatically correct sentences. By the late 1970's, IBM started building considerable models of the relationship between words and parts of speech [4] and by the early 2000s Bayesian statistics had been applied in the context of text modeling, whereby the topic probabilities provide an explicit representation of a document [5].

Behind these approaches, lay a belief that meaning could be extracted by the use of prior knowledge, either about the grammar and syntax of a language, or about topics in existence. As a consequence, much of the effort into mining text for meaning has

exploited the growing number of algorithms for supervised learning, such as Bayesian and Neural Networks and Support Vector Machines [6]. Primarily such techniques are used to categorize new texts against the categories of the training set, which in turn are in the gift of the supervisor. Moreover, cross-validation can be used to improve the performance of the model, albeit at the risk of under-generalisation and over-fitting.

There are however structural limitations in the supervised approach. Quite apart from the dangers of over-fitting by parameter tuning, there is a fundamental question about the codification of bias. What is the correct syntax of this sentence? What is the grammar and what is the dictionary? What decides the categories? For empiricists, the answer should be evident only from the texts under examination. That would however require machine learning with minimal supervision, as close as possible to unsupervised learning.

The experiments we describe in this paper expand on a notion first proposed by John R. Firth, Professor of General Linguistics at SOAS until 1956. Firth stressed the context dependent nature of meaning and is best known by the proposition that "You shall know a word by the company it keeps".

In brief, we apply the A-Priori algorithm to identify Frequent Item Sets (FISs) of words that occur together, and grade those sets by the singleton Association Rules (ARs) that each set implies. Singleton rules in this context contain just one term in the conclusion, as in this example from a Sports News source of a FIS {Hungarian, Grand, Prix}.

AR1: {Hungarian, Prix} $\Rightarrow$ Grand

the presence of the words "Hungarian" and "Prix" strongly imply the present of the word "Grand", within a segment of text

AR2: {Grand, Prix} $\Rightarrow$ Hungarian
AR3: {Grand, Hungarian} $\Rightarrow$ Prix

The literature on AR mining on text corpora does not constitute an overload, yet. There however have been some notable papers. An early approach by Maedche and Staab [7] mined ARs from frequent pairs, and built an ontology from the connection matrix. Our work extends it to FIS of arbitrary length, enabling ontologies with greater granularity to be discovered. Our approach does though focus on the single item conclusion in order to avoid the "powerset" problem described in section 2. At about the same time, Hristovski et al. [8] produced an interactive AR builder which generated ARs from a given start point. In particular, ARs were formed by associating the output of a search engine to the input query start point; the authors then went on to use logical inference to generate new propositions from those ARs, and human experts to prune the results. Given the user supplied start point and pruning, this approach is heavily supervised and contrary therefore to our minimal supervision requirement. It does however point out the additive benefits of rules in general, and Association Rules in particular. We also use this principle to accumulate a rule base which could also be open to logical inference.

More recently, Word2Vec [9] proposed an approach which recognises meaning through the "company the words keep". In Word2Vec, each word is associated with a

context derived from other words in its vicinity and encoded in a vector. This is achieved through a neural network learning approach. Similar words found in the same space will have a high similarity in their vectors as measured by the cosine distance. Such vectors are immediately usable by computers, and have proved extremely useful in machine translation, where similarity is a key criterion. That being said, the approach is not immune to traditional "Black Box" objections to neural networks; the output vectors have little meaning for humans, and so may not ultimately reduce information overload.

The area of topic discovery from multiple text streams (hence differently expressed) is highly relevant to our research. Wang et al. [10] claim that correlating multiple streams "provides more meaningful and comprehensive clues for topic mining". Their angle is to identify and bring together the same topics over long periods of time. Wang et al. [11] also agree on convergence of sources, but in contrast, focus on short periods of time. Here, they identify 'bursty" types of news, which is the same item intensively covered across multiple streams. Our approach shares the same objctives of identifying useful patterns in news. However, the novel feature is that we do not require a vocabulary of words as both the above papers do. This means that our approach is much more versatile to changing news agendas.

Finally, work in the detection of word collocations is fundamental to parsing, translation and summarization [12]. Collocations are frequent combinations of words that co-occur more often than expected by chance [13]. Examples of collocations include, compound proper nouns, compound nouns, noun-adjective combinations, proverbs and quotations. In our work, the words are those which identify a news story and which occur in a proximity to each other. These can also include compound proper nouns, such as peoples' names or countries. In collocation algorithms, once the text has been broken into n-grams, lookup and/or statistical frequency are deployed to isolate collocations from the text. There is similarity here in that our Association Rule approach also uses the frequency or "support" to propose a set of words. Out of choice, we do not use lookup in our approach. Therefore we may identify frequent item sets such as, {Romney, Trump, Mitt, Donald}, but identifying the collocated words within it, would impose an order of Mitt Romney and Donald Trump. This would come for us in a post-processing stage.

In the sections that immediately follow we describe the properties and challenges of Association Rules, and then move in section 3 to explain how our system "Sherlock", adapts this approach. Section 4 describes the results of an experiment to extract key topics from RSS newsfeeds and is followed by our conclusions from the work to date, and suggestions for ongoing work.

2 Association Rules and Frequent Item Sets

In formal terms, ARs are statements in the Propositional Calculus, which is a branch of Mathematical Logic concerned with deriving the truth or falsity of propositions derived from other propositions. It is the oldest form of logical manipulation, being attributed to Chrysippus in the third century BC. Correct use of its rules of inference remains a

test for arguments to this day, where it is known as Boolean algebra. A less formal point is that we are very accustomed to digesting complex information presented in this way. ARs extend the calculus by adding an empirical measure of interestingness to such statements, in order that validly inferred rules may be compared. They are used to discover correlations between variables, and drive the Market Basket Analysis we see at Amazon, which shows purchase patterns similar to a proposed purchase [14].

Behind every AR there must be a set of terms forming the premise, "if" part of the rule, and the terms that form the conclusion, "then", part of the rule. Such sets are called Frequent Item Sets (FIS). Identifying FIS is a combinational problem that involves exploring and counting the Lattice of variable combinations. Such a Lattice for just three variables a, b, and c is shown in figure 1. In order to show a property of all such lattices, namely that of downward closure, a set of length N has a maximum frequency of X, where X is minimum frequency found in its constituent sets of length N-1. Exploiting this property, efficient algorithms such as A-Priori can find all frequent item-sets.

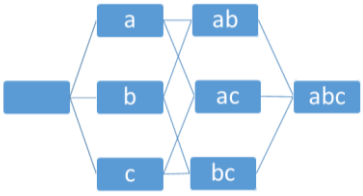


Figure 1: Lattice for Generating FIS

The exploration of the lattice of possible combinations can be conducted depth or breadth first, building either trees with counts for each node, or arrays expressing the same empirical evidence about the frequency of each set's constituents. Interested readers should visit Christian Borghelt's [15] repository of algorithms to see the range of approaches that have been adopted. Whatever the approach taken to the identification of FISs and their counts, two issues need resolution before statistically sound ARs can be extracted from the FISs identified. The first is the "powerset" problem and the second concerns the selection of the measure of "Interestingness".

The "powerset" problem is that any member of the powerset of a FIS, except both the full and empty sets, could be the premise set of an AR, so a FIS of length 3, as shown above implies (2^N) -2 possible premise sets, and therefore (2^N) -2 rules. In this case one FIS implies 6 rules, each of which need counting. Possible premise sets are, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}. The longest FIS found in our Newsfeed experiment has 11 members. This means $2^{11} - 2$, or 2046 different ARs, all of which need to be counted. The issue is therefore one of combinatorial explosion of FIS which need to be examined.

The second issue that needs to be addressed is the selection of an appropriate measure of "Interestingness". In their paper, Pang-Ning et al. [16] demonstrate that all measures of Interestingness have biases such that rules may be graded differently by different measures. We commend that paper to the reader, along with a comprehensive collection of links to measures of Interestingness maintained by Hahsler (2016).

3 Finding the FIS and ARs

Sherlock uses the A-Priori algorithm to generate FISs, restricts rule generation to a single term conclusions in order to avoid the "powerset" problem, and uses Iterative Proportional Fitting (IPF) to produce the most objective measure of Interestingness (Pang-Ning et al, 2004). Sherlock therefore uses IPF to adjust the row and column totals of the observed data towards the balanced totals that would occur if the chances of the premise being true were 50%, and the chances of the conclusion being true were also 50%. By standardising the contingency table of each rule in this way we are able, albeit at further computational cost, to avoid introducing bias associated with any scoring system. The A-Priori algorithm generates FISs in a breadth first search, the set of all pairs is generated from the set of all sufficiently frequent single items, the set of all of all triples is generated from the set of all pairs, and so on until no further generation is possible.

The question that Sherlock seeks to answer is, what are the most important N variables (words) in this data (news) and how are they connected?

In order to avoid overwhelming available memory, the program operates by reducing the frequency threshold until the set of ARs, sorted by IPF adjusted score, contains at least N unique items. Using the rules derived from the data we can construct a network of the most important terms to demonstrate the connections, as in Fig. 1.

The data structure produced is a dynamic array in which each row represents a singleton AR; that is a rule with only one item in the conclusion, along with its score, where 0.5 is the maximum, due to iterative proportional fitting. It grows as data comes in; it starts with frequent items, it then appends pairs that also exceed the frequency threshold, from these are generated candidate triples, triples that get over the frequency threshold get appended to the array, and so on, towards larger sets.

Table 1 shows a typical output, starting with the last frequent items and the first 2 pairs. Therefore, for the 1st row, this singleton AR is derived from the 1st FIS of size one and has been seen 13 times already. Rows 3 and 4 constitute two rules from the same FIS of size two. Furthermore, we can see that the first rule derived from the second pair {rows 5 and 6} had the highest score for a singleton rule (0.37998). Note that frequent item sets of size 1 do not constitute singleton rules, as the premise would have to be an empty set.

Table 1. Snapshot of Singleton Rules

Row	FIS ID	Position	Set Size	Item ID	Set Count	Premise Count	Conclusion Count	Interest Score
1	1	1	1	23424	13	13	13	0.5
2	1	1	1	23457	14	14	14	0.5
3	0	0	2	16	6	58	16	0.36536
4	0	1	2	2272	6	16	58	0.36531
5	1	0	2	16	7	57	16	0.37998
6	1	1	2	10730	7	16	57	0.37982

In Table 2 we see that the table ends with eight quintuples, the last being made from the FIS {5798,8563,19708,7708,17210} which occurred 6 times and that the best rule produced by that set was rule number 245958 namely, {5798,8563,19708,7708} => {17210}, which had the highest score. This states that whenever the premise occurred in the data 17210 occurred nearly 10 times as often as the set.

Table 2. Long Association Rule

Row	FIS ID	Position	Set Size	Item ID	Set Count	Premise Count	Conclusion Count	Interest Score
245954	7	0	5	5798	6	8	61	0.42815
245955	7	1	5	8563	6	6	65	0.49549
245956	7	2	5	19078	6	7	39	0.45877
245957	7	3	5	7708	6	7	46	0.45468
245958	7	4	5	17210	6	6	56	0.49557

4 Data Processing

Using Newsblur.com, we established a pipeline of news which would append new stories from appropriate feeds defined in Newsblur to an input file in the Cloud. The news channels watched by Newsblur include,

- www.econmist.com/rss/the_world_this_week_rss.xml
- www.telegraph.co.uk/news/worldnews/rss
- rss.cnn.com/rss/edition_world.rss
- feeds.bbci.co.uk/news/rss.xml? edition=int
- feeds.skynews.com/feeds/rss/world.xml

Sherlock first converts each news item into a vector of interesting words, after stripping out tags and stopwords. These are words which do not carry significant meaning from the text, such as ‘and’ and ‘or’. If they were left in, then vast amounts of superfluous rules would be generated. Currently Sherlock supports English, Russian and Arabic texts, but there are no reason this cannot be extended to all languages that have separators to distinguish words. Sherlock then converts to binary the occurrence matrix into an integer vector against which counting can be performed in parallel, and repeatedly reduces the frequency threshold in the manner described above until the sorted singleton rule list that is returned contains at least 12 unique terms. In practice, due to the breadth-first nature of the search, the returned rule set can jump to exceed the minimum term threshold. In the screenshot shown in Fig 2 we see that when the

frequency threshold was reduced to 1.4% 20 unique items were in the results, in 6 distinct clusters. The diagram can be interpreted from the point of view of a node having one or more links into it. This node is the Conclusion and the nodes pointing to it are the premise. Therefore, we have from Figure 2 we have both, {Nepal, Prince} ⇒ {Harry}, {Nepal, Harry} ⇒ {Prince} – two rules referring to a story about Prince Harry visiting Nepal, but compressed into one cluster.

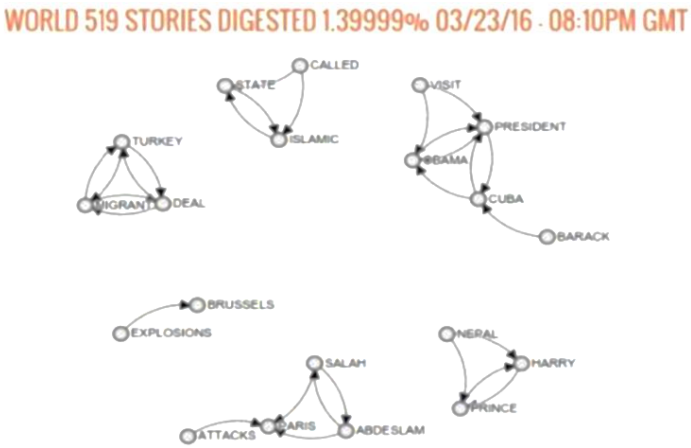


Figure 2: Graphical Illustration of ARs forming Clusters (Rule Sets)

5 Results

Over the period Feb 2015 to March 2016 more than 32,000 updates from news sources have been received. Each causes Sherlock to check for existing ARs and to record findings in a database, along with a summary report. The 227,515 FIS to March 26 are available for download at http://www.meme-machines.com/world4.csv.

963 unique FISs were identified, ranging in length from 2 to 11 according to the following distribution in Table 3.

Table 3. Distribution of FIPs in terms of Number of Items

FIS Length	Frequency
11	1
7	4
6	7
5	24
4	119
3	439
2	369
Total	963

Due to the nature of the A-Priori algorithm, there are many cases of a FIS being a subset of a longer FIS. These subsets are equally plausible, but contain less information about the news item. For example, taking as a starting the longest rule (11), we have found the following FIP decomposition which is also "interesting".

['Airport', 'Atlantic', 'Beam', 'Flight', 'Heathrow', 'Incident', 'Laser', 'London', 'Plane', 'Virgin', 'York']
['Airport', 'Atlantic', 'Beam', 'Flight', 'Heathrow', 'Laser', 'Virgin']

We can also find examples where a FIS is completely constructed from other subset FISs,

['Apple', 'Bernardino', 'FBI', 'San']
['Apple', 'FBI']
['Bernardino', 'San']

Subsets that remain after the subtraction of identified FISs also offer areas of potential interest.

['Airport', 'Atlantic', 'Beam', 'Flight', 'Heathrow', 'Incident', 'Laser', 'London', 'Plane', 'Virgin', 'York']
['Airport', 'Atlantic', 'Beam', 'Flight', 'Heathrow', 'Laser', 'Virgin']

suggests ['Incident', 'London', 'Plane', 'York'], while

['Donald', 'Mitt', 'Presidential', 'Republican', 'Romney', 'Trump']
['Donald', 'Mitt', 'Romney', 'Trump']

suggests ['Presidential', 'Republican'].

Some sets are not decomposable to the smallest basic length of 2. We contrast [Amanda, Knox, Murder], which does decomposes to [Amanda Knox], with [Demi, Drowned, Man, Moore, Pool] which does not decompose further. All this follows from how the news stories are written by the various news sources and their development over time. With filtering in place, only news stories covered frequently will emerge. Further, if they are reported in a consistent way they will more likely appear. With proper compound nouns, such as name or places these are unlikely to be changed from one news item to another. The event surrounding the name may though be described differently.

6 Conclusions and Future Work

With no knowledge of the domain, the grammar, or the language (subject to constraints about stopwords), the derived Association Rules describe meaningful major and minor pieces of news using logic and frequent words. Even with multiple sources of data, news items do use the same set of words to identify topics. Information overload, as we proposed at the beginning, can be reduced automatically to coherent pieces, pointing towards more detail descriptions through Internet search. The results come directly

from the combination of news reporter articles and the A-Priori algorithm. As such, the results reflect how a news item may be written to have a dominant theme (longest item set), while also going on to develop the article through subthemes, evidenced by subsets of FIS. Subtracting item sets from FISs suggest likely candidates for shorter FISs or topics for the reporter. The automatic identification of news themes though gives a way of indexing them for easier retrieval later. The time dimension, although not explored here, can give an insight into the life cycle of a news item; when it is reported and its strength of reporting over time.

Sherlock is available and can be validated at www.meme-machines.com

References

1. Diderot, D., le Rond d'Alembert, J.: Encyclopédie, Vol. V. Paris (1756)
2. Toffler, A.: Future Shock, Bantam, Books, New York (1970)
3. Evaristo, R., Adams, C., Curley, S.: Information Load Revisited: A Theoretical Model. In: Proceedings of the 16th Annual International Conference on Information Systems, Amsterdam, pp. 197-206 (1965)
4. Bahl, L. R.., Baker, J. K., Cohen, P. S., Jelinek, F., Lewis, B. L., Mercer, R. L.: Recognition of Continuously Read Natural Corpus. In Proc. IEEE Int. Conference in Acoustic Speech and Signal processing, Tulsa, OK, US, pp 422-424 (1978)
5. Blei, D. M., Ng, A. Y., Jordan, M. I.: Latent Dirichlet Allocation. The Journal of Machine Learning Research, Vol. 3, pp 993-1022 (2003)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment Classification using Machine Learning Techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10(pp. 79-86). Association for Computational Linguistics (2002)
7. Maedche, A., Staab. S.: Discovering Conceptual Relations from Text. In Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'2000). Vol. 321, no. 325. Ed. Horn, W (2000)
8. Hristovskia, D., Starea, J., Peterlinb, B., Dzeroskic, S.: Supporting Discovery in Medicine by Association Rule Mining, in Medline and UMLS. Medinfo. Vol. 10, pp 1344–1348 (2001)
9. Mikolov. T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR (2013)
10. Wang, X., Zhang, K., Jin, X., & Shen, D.: Mining Common Topics from Multiple Asynchronous Text Streams. In Proceedings of the Second ACM International Conference on Web Search and Data Mining pp 192-201 (2009)
11. Wang, X., Zhai, C., Hu, X., Sproat, R.: Mining Correlated Bursty Topic Patterns from Coordinated Text Streams. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining pp 784-793 (2007)
12. Verma , R., Vuppuluri, V., Nguyen, A., Mukherjee, A., Mammar, G., Baki, S., Armstron, R.: Mining the Web for Collocations: IR Models of Term Associations. In Proceedings of 17th International Conference on Intelligent Text Processing and Computational Linguistics (2016)
13. Smadja, F.: Retrieving Collocations from Text: Xtract. Computational Linguistics Vol 19, pp 143–177 (1993)

14. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I.: Fast Discovery of Association Rules. Advances in knowledge discovery and data mining, 12(1), pp 307-328 (1996)
15. Hahsler. M.: A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules, http://michael.hahsler.net/research/association_rules/measures.html (2016)
16. Pang-Ning, T., Vipin, K., Jaideep, S.: Selecting the Right Objective Measure for Association Analysis, Information Systems. Vol. 29, no. 4, pp 293-313 (2004)

Detecting Communities Using Link and Content Triangles

Qiuling Yan[1], Baoli Li[2], Dongqing Yang[1]

[1] Department of Computer Science, Peking University, Beijing, China
[2] Department of Computer Science, Henan University of Technology, Zhengzhou, China

yqlpku@gmail.com,dqyang@pku.edu.cn, csblli@gmail.com

Abstract. Community detection for uncovering the hidden community structure in large networks is an important task in analyzing complex networks. Most of the existing methods only consider link structure in networks, where the link information is usually sparse and noisy, which may result in a poor partition of a network. Fortunately, besides link structure, nodes, especially in social networks, are often associated with certain symbolic or textual attributes, which we refer to as content. Content, therefore, is expected to serve as a reasonable complement for finding a good partition. In this work, we propose an algorithm LICT to detect communities with link and content triangles. It works in three steps: 1) network expansion with content similarity; 2) community detection in weighted network; and 3) refinement by weighted triangle modularity. Experimental results on several real data sets demonstrate that the proposed algorithm is effective for community detection and robust in the presence of link noise.

Keywords: community detection; social network analysis; link and content triangles; weighted triangle modularity; spectral optimization

1 Introduction

Real networks are often organized in local clusters called communities, which can be considered as relatively independent modules. Nodes in the same community are more densely connected to each other than that of nodes in different communities. Communities can occur in many networked systems. For example, in social networks, a community is a group of friends that communicate with each other much frequently. In citation networks, a community is a set of papers that have citation relationship and focus on the same topic. In protein-protein interaction networks, communities are a group of proteins having the same specific function within the cell. Thus Detecting communities is crucial to understand the structural properties of networks [1] and helpful to improve other tasks such as link prediction [2].

Many existing methods only use network structure to detect communities. However, there exists noise in networks, representing as incorrect links and missing links, which weaken clustering quality. To reduce the impact of noise, content is a good complement. The similarities and differences in the content of nodes can affect the patterns of

linking. Thus, it is sensible to combine links and content together to detect communities. There exist some solutions aiming at this problem, which can be categorized into two classes. One is generative probabilistic modeling [3] [4] [5] [6] [7] [8] [9]. Although these solutions can model links and content simultaneously, they are either too complex to be applied or only able to handle relatively small networks. Another type of approaches is heuristic[10] [11] [12] [13]. They either embed content information into edges or store link structure into a distance function between nodes. However, these methods either limit content to attributes of nodes or lose the ability to discriminate different nodes when too many features of content are involved.

In this work, we propose a simple but effective algorithm to detect communities using link and content triangles. It works in three steps. First, given a network, we add new edges into the network according to content similarity. Then we compute weights of edges using both structural information and content similarity. At the second step, we use *k*-way spectral method to partition the weighted network. Thirdly, the partition is refined according to weighted triangle modularity. We apply the method to several real networks. Experimental results show that it is effective for community detection and robust in the presence of link noise.

The paper is organized as follows: section 2 presents related work; section 3 explains the proposed algorithm; section 4 shows the experimental results, and section 5 concludes the paper.

2 Related Work

A lot of algorithms have been proposed in the past years to detect communities in a complex network. Fortunato provides an excellent survey[14]. Here, we focus on related work in two specific directions, as they are highly relevant to our proposed algorithm. One is to combine links and content to detect communities, and the other is the usage of triangles in network analysis.

Community detection using both links and content: There are various approaches to utilize both sources, which can be categorized into two classes. One class is generative probabilistic modeling [3] [4] [5] [6] [7] [8] . In these works, it was assumed that either community generates links and content or communities and content generate link structure. For example, Liu et al. [3] argued that network structure is dependent on both communities and content. The authors in [4] merged the idea of topic model and stochastic model, with the assumption that links and content share the same topic space. Sun et al. [5] proposed a probabilistic model that clusters the objects of different types into a common hidden space. Nallapati et al. [8] used LDA and PLSA to model citing documents and cited documents respectively and introduced a method called Link-PLSA-LDA to jointly model content and links. Similar to [3] [4] [8], topic model based approaches are also proposed in [6] [7] [9].

Another popular category to combine links and content is the hybrid approaches [10] [11] [12] [13], most of which computes pairwise distances by fusing similarities of links and content. Akoglu et al. [10] proposed a method that compresses adjcent matrix and

feature matrix simultaneously to disclose community blocks. Ruan et al. [11] constructed content edges and fused them into original network to get an extended network with the same vertices. Then he sampled the network to obtain a sparse one and applied some existing methods to partition the sampled network. Zhou et al. [12] introduced a method named as SA-Cluster which inserts attribute nodes to a network to get an augmented network. Then they used the neighborhood random walk model to estimate the vertex closeness on the new network. Moser et al. [13] integrated the concepts of dense subnetworks and of subspace clusters in a feature space. Then they find out subsets of nodes that are close in the feature space. Our work is inspired by the work [11]. The difference is that we convert pairwise similarity into edge weights and use triangles modularity to improve partition quality.

Community detection based on triangles: Since many metrics in network analysis can be obtained by graph triangulation, it provides insight into social network analysis[15] [16] [17]. Coefficient and transitivity are representative, which are two important metrics quantifying density of sub-networks. Consequently, we can use triangles to improve community results. For example, Klymko et al. [18] applies triangles information to detect community in directed networks. Prat-Pérez et al. [19] assumes that well-defined communities are dense in terms of triangles. Accordingly, he proposed a metric called WCC to measure the quality of community results. Serrour et al. [20] extends the modularity metric with triangles. The most prominent difference between our work and the works above is that we utilize content information as well as structure.

3 Community Detection Using Link and Content Triangles

Let $G(V, E, T)$ be an undirected network. V is the set of vertices $(v_1, v_2, \ldots, v_n)$. E is the edge set without weights. Each node v_i in V corresponds to a content vector t_i in T. Our goal is to cluster vertices according to both network structure and content similarity, with the assumption that the density of triangles in a cluster is larger than that outside the cluster. In this work, we propose a method called LICT[1], which consists of three steps. First, we add edges and weights to the original network, according to link structure and content similarity. Then any state-of-art method that aims at weighted networks can be applied on the new network. At the last step, we refine the partition according to weighted triangle modularity, a metric that accords with our intuition that triangles are building blocks for community.

Now we proceed with more details.

3.1 Combining Links and Content

To combine links and content, our idea is to compute pairwise affinity of vertices utilizing both link structure and content similarity. Then we add weights and some

[1] Detecting communities using **LI**nks and **C**ontent based on **T**riangles.

edges to the original network to get a weighted one. Algorithm 1 demonstrates our idea in detail. For each node pair (v_i, v_j), we compute the cosine similarity between vector t_i and t_j (Line 2-4). Elements in each vector t_i can be a binary value, or TF-IDF value, or number of word occurrence. After computing pairwise content similarity, for each node *v* we choose top *K* vertices and add edges between *v* and those vertices into the original network (Line 5-7) to get a new network *G*'. To avoid over-expanding network, we do not consider those similarities smaller than a threshold *T*. Consequently, there exists the case that the number of new neighbors for a node is smaller than *K*. In addition, if there already exists an edge between node *v* and one of its top *K* vertices, we do not need to add a new one. In line 6, Neighbors(*v*) is the neighbor set of node *v*. To decide the value of *K*, the scale of a network is an indispensable factor to take into account.

Algorithm 1: Converting a network to a weighted one
Input: a network $G(V, E, T)$ without weights Output: a weighted network $G'(V, E')$
1. $G' \leftarrow G$ 2. For each pair (v_i, v_j) in G' 3. Compute content similarity; 4. End For 5. For each v in V 6. Add top K vertices to Neighbors(v) according to content similarity and add edges to G' accordingly; 7. End For 8. For each edge (u, v) in G' 9. Weight of edge(u, v)$\leftarrow \partial \times simt_{uv} + (1 - \partial) \times simc_{uv}$ 10. End For

Fig. 1. Combining links and content

At line 9, we combine link structure and content similarity to compute weights for edges in *G*'. $simt_{uv}$ represents structural affinity for node *u* and node *v* in the original network *G*, which is computed as $simt_{uv} = \frac{1}{l_{sp}}$, where l_{sp} is the length of the shortest path between *u* and *v* in network *G*. Bidirectional search algorithm is used to compute $simt_{uv}$. For a node *v*, since l_{sp} is known as 1 between *v* and one of its original neighbors in *G*, we only need to compute length of the shortest path between *v* and its new neighbors. $simc_{uv}$ is the value of content similarity between *u* and *v*, which is normalized using *zero-one* in globe scope. ∂ is a balancing coefficient between $simt_{uv}$ and $simc_{uv}$.

It is important to note that our method can be also extended to weighted networks, by merging the new weights with the original ones into the networks.

By now, we convert a network $G(V, E, T)$ without weights to a weighted network $G'(V, E')$. Then any state-of-art method that aims at weighted networks can be applied on the network $G'(V,E')$ to get a partition of *V*. In this work, we use *k*-way spectral

clustering method [21]. After clustering the vertices, we need to refine the partition results. Our refinement method is described in next section.

3.2 Refining Clustering Results

In a network $G(V, E)$, a triangle is a complete subnetwork that consists three nodes $(u, v, w) \in V$ and three edges $\{(u, v), (v, w), (u, w)\} \in E$. Triangles play an important role in network analysis. Many metrics of networks can be computed directly by counting triangles, such as cluster coefficient [22], neighborhood density [15]. Triangles are also useful to improve clustering quality [18] [19] [20] [23]. In our work, we assume that density of triangles inside a community is larger than that across different communities. Then we propose a metric called weighted triangle modularity and use it to refine the initial partition obtained in previous section.

Weighted triangle modularity is an extension of modularity[24], which is a widely used metric. The modularity metric is based on the assumption that there are more dense edges in a community than in a random network with the same degree distribution. Given a partition $P=\{C_1, C_2, \dots, C_k\}$ of a network G, the generalized definition of modularity is as follows.

$$Q(P) = \frac{1}{2w}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(w_{ij} - \frac{w_i^{out}w_j^{in}}{2w}\right)\delta(C_i, C_j)\ . \quad (1)$$

Where w_{ij} is the weight of an edge (v_i, v_j). If there is no edge between v_i and v_j, w_{ij} is zero. $w_i^{out}(=\sum_j w_{ij})$ is the degree going from the node v_i, while $w_j^{in}(=\sum_i w_{ij})$ is the strength of links coming to the node v_j. C_i is the index of a community to which node v_i belongs. Finally, $\delta(C_i, C_j)$ is the Kronecker function assigning to 1 if node v_i and node v_j belong to the same community, 0 otherwise. The larger the Q value is, the better the community partition is.

Through comparing density of triangles rather than density of edges, we extend modularity to get weighted triangle modularity. The formula of this metric is as follows.

$$Q(P) = \sum_{i,j,k,} B_{ijk}\delta(C_i, C_j)\delta(C_j, C_k)\delta(C_k, C_i)\ . \quad (2)$$

With the conditions that $(i,j),(j,k),(k,i) \in E$ and triangle inequality holds among w_{ij}, w_{jk} and w_{ki}.

In equation 2, B_{ijk} is the mathematical object that evaluates difference of the triangle density between a sub-network and a corresponding random network. We compute B_{ijk} according to the following formula.

$$B_{ijk} = \frac{1}{T_G}w_{ij}w_{jk}w_{ki} - \frac{1}{T_R}(w_i w_j)(w_j w_k)(w_k w_i)\ . \quad (3)$$

T_G is the total number of triads of nodes that form triangles in the network G. The formula of T_G is:

$$T_G = \sum_i \sum_j \sum_k w_{ij}w_{jk}w_{ki}\ . \quad (4)$$

T_R is the counterpart of T_G in null case. The formula for T_R is as follows.

$$T_R = \sum_i \sum_j \sum_k (w_i w_j)(w_j w_k)(w_k w_i) \ . \quad (5)$$

In equations 3-5, w_i is the sum of weights for edges that node v_i intervenes.

Although Serrour [20] also extended the modularity using triangles, he did not control conditions. We believe that those conditions are important to detect more cohesive communities.

With weighted triangle modularity, we further refine the initial partition obtained in previous section. The heuristic idea is to move vertices among communities to increase the value of weighted triangle modularity. This process is repeated until Q does not increase any more. We demonstrate the details in algorithm 2. To be noted, we use the network obtained using algorithm 1 rather than original network in this phase.

When moving node *v* from a community to another, it is sensible to choose the communities to which node *v* connect densely, rather than trying each of other communities (Line 4). Triangle number that node *v* involves in a community can be set as the choosing criterion.

In algorithm 2, the computation for the Q value costs mostly. Since the computation part of Q that does not relate to node *v* stay unchanged, we only need to focus on the relative change of Q, which relates to two communities at most: the source community C_s and the destiny one C_d. Let us consider the simplest case firstly that the source community C_s only contain node *v*. when moving node *v* from C_s to C_d, the relative change of Q is computed as follows.

$$\Delta Q_I = \sum_{i,j,k \in C'} B_{ijk} - \sum_{i,j,k \in C_d} B_{ijk}, \quad (6)$$

where $C' = C_d \cup \{v\}$.

Since the relative change ΔQ_I results from participation of node *v* into community C_d, we only need to consider triads involving node *v* in community C_d. Thus we can rewrite equation 6 as follows.

$$\Delta Q_I = \sum_{i,j \in C_d} \left(\frac{1}{T_G} w_{vj} w_{ij} w_{jv} - \frac{1}{T_R} (w_v w_i)\left(w_i w_j\right)\left(w_j w_v\right)\right) \ . \quad (7)$$

With the conditions that $(v,j),(j,v),(i,j) \in E$ and triangle inequality holds among w_{vj}, w_{jv} and w_{ij}.

Now suppose that the source community C_s for node *v* contains other nodes. We have the following theorem.

Theorem 1. Let $P = \{C_1, C_2, \dots, C_s, C_d\}$ and $P' = \{C_1, C_2, \dots, C_s', C_d'\}$ be two partition for network *G*(*V*, *E*), where $C_s' = C_s \backslash \{v\}$, $C_d' = C_d \cup \{v\}$. Then, when moving node *v* from C_s to C_d, the change of Q is computed as follows.

$Q(P') - Q(P) = -\Delta Q_{Is} + \Delta Q_{Id}$. (8)Where ΔQ_{Is} is the change for node *v* from the community $\{v\}$ to be inserted into community C_s, which can be obtained using equation 7. ΔQ_{Id} is the same case as ΔQ_{Is} except that the destination is community C_d.

Proof. Moving node *v* from C_s to C_d can be considered as two steps, each of which leads to a new partition of $G(V, E)$. For each step, we use Q_{old} and Q_{new} to represent Q for the old partition and the new one, respectively.

In the first step, node *v* is removed from C_s to get a community consisting only of node *v*. In the second step, node *v* is inserted from the community $\{v\}$ to community C_d. As for the first step, suppose that ΔQ_R is the change of Q, the following formula holds obviously.

$$Q_{old} = Q_{new} + \Delta Q_{Is},$$

Then $\Delta Q_R = Q_{new} - Q_{old} = -\Delta Q_{Is}$. Since the change of Q in the second step is ΔQ_{Id}, equation 8 holds by combining the two steps.

Algorithm 2: Refinement

Input: partition P of a weighted network $G'(V, E')$
Output: refinement partition P'

```
1. P' ← P;
2. Repeat
3.     For each v in V
4.         candidates ← candidateComm(v, P');
5.         For each C in candidates
6.             Compute ΔQ(v, SourceC, C) according to equation 8;
7.         End For
8.         Choose the candidate C* that maximizes ΔQ value;
9.         If ΔQ > 0, move v from SourceC to C*;
10.    End For
11. Until Q value does not increase
```

Fig. 2. The process of refinement

3.3 Complexity of Algorithms

Given a network $G(V, E, T)$, let *n* be the number of nodes and *m* the number of edges. We assume that the average degree in *G* is $d=2m/n$.

In algorithm 1, the costly part is to compute pairwise similarity of content among vertices (Lines 2-4). The complexity for this part is $O(tn^2)$, where *t* is the length of content vector. The loop in lines 5-7 is $O(nK)$, where *K* is a constant given by users to choose top *K* new neighbors. To compute structural affinity in line 9, the complexity is $O(b^{l/2})$ for a pair of nodes (u, v) in G', where *b* is a branching factor and *l* is the length of the shortest path. Thus, for the loop in lines 8-10, the complexity is $O(nKb^{l/2})$. Then the complexity of algorithm 1 is $O(tn^2 + nK + nKb^{l/2})$. For a network that $Kb^{l/2} \ll n$, the complexity is $O(tn^2)$. It means that computing pairwise similarity of content takes the most time in algorithm 1.

Remember that we adopt *k*-way spectral optimization to get an initial partition, the main cost results from *k*-means algorithm. Since the cost of other part can be negligible, the complexity for this phase is $O(nka)$, where a is the number of iteration for *k*-means.

Now we turn to algorithm 2, the refinement part. To move a node *v* from a community to another one, we need to compute the change of Q value. The corresponding cost

is $O(c^2)$, where c is the average scale of communities in a partition. In fact, to choose candidate communities, we can adopt some heuristic schemes, rather than examining every community. For example, voting is a feasible choice, which can reduce the cost to O(1). Then the computation for loops in line 3-10 is $O(nd^2)=O(m^2/n)$. Let r be the number of iteration to find the best partition, the total complexity for algorithm 2 is $O(rm^2/n)$.

To sum up, the total cost for our method is $O(tn^2+ nka+ rm^2/n)= O(tn^2)$. In another word, computing content similarity costs mostly in this work.

4 Experiments

4.1 Datasets

In our experiments, we use three real datasets with different domains ranging from citation networks to social networks, all of which are treated as undirected. Each dataset is described below.

- CORA[2][25]. This is a citation network, in which each paper is considered as a node. We randomly choose a seed and use breadth-first search to get a small network with 2527 papers and 8427 edges. The small network is used as our first dataset. We extract title and abstract as content for each paper, which is represented as a vector of word occurrence. Our dictionary contains 5688 words. Each paper is labeled with a category. There are 10 classes for the chosen papers, which is defined as ground-truth communities.
- Flickr. We use the dataset used in the work [11] as our second dataset, which was gathered from the Flickr site. This is a user-user contacting network, which contains 16710 users and 716,063 edges among users. Tags adopted by users for photos are used as content information. The elements of content vectors are binary. There are 184421 user groups and a user can join in several groups. We use these groups as ground-truth communities.
- Facebook[3]. The dataset includes several ego-networks, consisting of 4039 nodes and 88234 edges. User profiles are used as content information, including locations, education information and so on. The social circles are labeled by the owners of ego-networks. We use those social circles as ground-truth communities.

4.2 Experimental Settings

In algorithm 1, we need to decide the parameter K for choosing top K content neighbors for each node. Since we extend original networks by adding edges according to content similarity, we assume that the number of new edges is no more than original edges. Specifically, we set K as 10, 50 for Cora and Flickr, respectively. We set K as 5 for Facebook because that the average size for each ego-network is small. Besides, to filter

[2] https://people.cs.umass.edu/~mccallum/data.html
[3] http://snap.stanford.edu/

content similarities, we set the threshold T as the average of content similarities in each network. In algorithm 2, parameter ∂ balances the contributions of structural information and content similarity. Since we are interested in how different ∂ influences performance of the whole method, we would like to examine different values for ∂. When using k-way spectral method to get an initial partition for a network, we set the community number k as the same with that of ground truth, except the Flickr dataset. Different from CORA and Facebook, a node in the Flickr dataset can belong to several communities. For simplicity, we set k as 50 for the Flickr dataset.

We choose two other methods as the baselines. One is the method proposed in this work, without regard to content. It enables us to examine to which extent content information contributes to community detection. We call this method LIT[4] for the sake of convenience. In the process of refinement for LIT, we set the weight as 1 for an edge. The other baseline method is CODICIL[11], which detects communities using links and content. Comparing our method LICT with CODICIL can help us to investigate the role of triangles in community detection. We set parameters of CODICIL in the same way as LICT, and also use k-way spectral method for CODICIL. Among state of art methods that combine links and content, SA-Cluster[12] and Link-PLSA-LDA are typical. The former is heuristic and the latter is a generative probabilistic model. Since CODICIL has been shown to outperform the two methods, we do not compare LICT with them.

Given a predicted partition P and the ground truth P' for a network G, we compute average F1-score used in [26] to measure the clustering quality. Specifically, both the predicted communities and ground-truth communities are considered as reference. After matching predicted communities with those in ground-truth sets, we also match ground-truth communities with predicted ones. Then the performance is measured by the average of F1-scores, which is calculated as follows:

$$F1(P,P') = \frac{1}{2|P|}\sum_{C_i \in P} F1(C_i,P') + \frac{1}{2|P'|}\sum_{C_i \in P'} F1(C_i,P)$$

$$F1(C,P) = \operatorname{argmax} F1(C,S_i), S_i \in P = \{S_1, \dots, S_n\}$$

$$F1(C,S) = \frac{2\times p\times r}{p+r}, p = \frac{|C\cap S|}{|C|}, r = \frac{|C\cap S|}{|S|}.$$

4.3 Experimental Results

First of all, we examine the clustering quality for the proposed method compared with baselines. For the method LICT and CODICIL, we set parameter ∂ as 0.6, unless noted otherwise. We show the result in Figure 3.

Compared with CODICIL, the proposed method LICT performs better on all of three datasets. Although both LICT and CODICIL leverage links and content, LICT uses triangles in community detection. The results verify our intuition that triangles play a role for the improvement of clustering quality. On the other hand, LICT outperforms LIT in all cases, which shows that content information is valuable for clustering. Besides, LIT performs better than CODICIL on Facebook. The reason is that the Facebook dataset is ego-networks, which contains more triangles than other networks. This enables LIT to work well, in spite of ignoring content. No matter what domain of a network is, LICT can be applied more widely than both LIT and CODICIL.

[4] Detecting communities using **LI**nk **T**riangles.

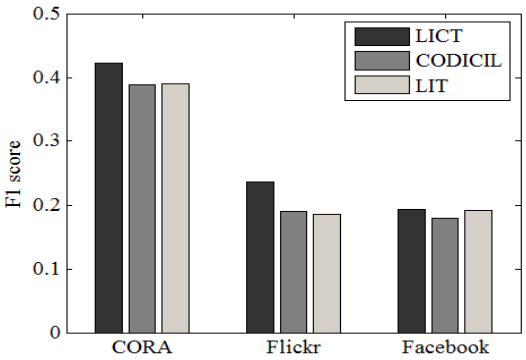


Fig. 3. Performance comparison of LICT, CODICIL and LIT in term of F1 score

To investigate the impact of parameter ∂, we apply LICT on the three datasets with ∂ valued from 0.1 to 0.9, stepping by 0.1. We show the results in Figure 4. For CORA and Flickr, we get the highest F1 scores with $\partial = 0.6$, so we set $\partial = 0.6$ in the rest of experiments.

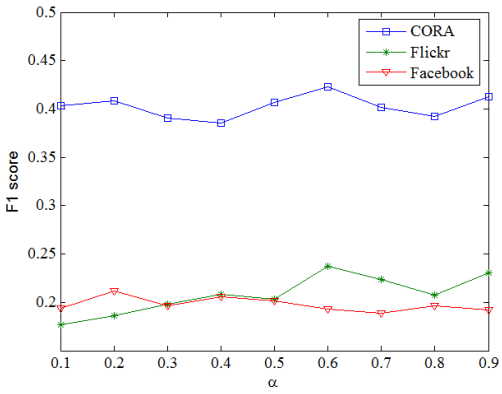


Fig. 4. Performance of LICT with ∂ valued from 0.1 to 0.9

To further investigate the role of content information, we remove some fraction of edges randomly and apply both LICT and LIT to the obtained networks. Figure 5 shows the relative improvement of LICT compared to LIT. For all of three datasets, when we remove more edges, relative improvements of LICT increase. Especially for the document network CORA, the improvement is much more obvious. Thus, when the network becomes unreliable or contains link noise, we can use content information to improve clustering quality.

5 Conclusion

In this work, we propose a method that combines links and content to detect communities. The method consists of three steps. First, we add edges to a network according to

content similarity and convert the network to a weighted one. Then we apply *k*-way spectral algorithm to get an initial partition for the weighted network. In the third step, we refine the partition further according to weighted triangle modularity. Experimental results on several real datasets show that the proposed method is effective for detecting communities and robust in the presence of network noise.

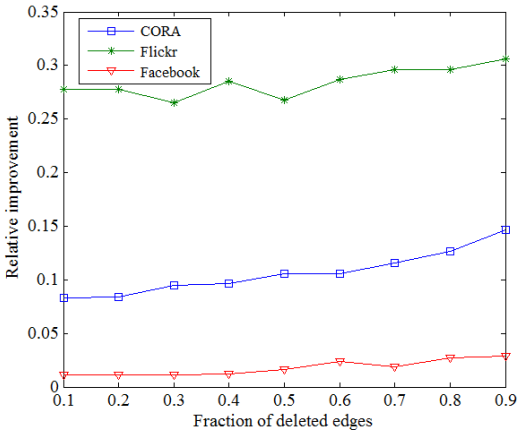


Fig. 5. Relative improvement of LICT against LIT when deleting edges

In the future, we plan to improve the work from two directions. Firstly, as the two procedures of choosing the top *K* nodes and refining partition with weighted triangle modularity are time-consuming, we would like to explore how to speed up these two parts. Secondly, we also consider how to make the proposed method applicable for much larger networks.

Acknowledgements

Baoli Li was partly supported by the Henan Provincial Research Program on Fundamental and Cutting-Edge Technologies (No. 112300410007), and the High-level Talent Foundation of Henan University of Technology (No. 2012BS027).

References

1. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U.: Complex networks: Structure and dynamics. Physics reports 424, 175-308 (2006)
2. Chang, J., Blei, D.M.: Relational topic models for document networks. In: International Conference on Artificial Intelligence and Statistics, pp. 81-88. (2009)
3. Liu, Y., Niculescu-Mizil, A., Gryc, W.: Topic-link LDA: joint models of topic and author community. In: proceedings of the 26th annual international conference on machine learning, pp. 665-672. ACM, (2009)
4. Balasubramanyan, R., Cohen, W.W.: Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In: SDM, pp. 450-461. SIAM, (2011)

5. Sun, Y., Aggarwal, C.C., Han, J.: Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. Proceedings of the VLDB Endowment 5, 394-405 (2012)
6. Xu, Z., Ke, Y., Wang, Y., Cheng, H., Cheng, J.: A model-based approach to attributed graph clustering. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 505-516. ACM, (2012)
7. Zhu, Y., Yan, X., Getoor, L., Moore, C.: Scalable text and link analysis with mixed-topic link models. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 473-481. ACM, (2013)
8. Nallapati, R.M., Ahmed, A., Xing, E.P., Cohen, W.W.: Joint latent topic models for text and citations. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 542-550. ACM, (2008)
9. Yang, J., McAuley, J., Leskovec, J.: Community detection in networks with node attributes. In: Data Mining (ICDM), 2013 IEEE 13th International Conference on, pp. 1151-1156. IEEE, (2013)
10. Akoglu, L., Tong, H., Meeder, B., Faloutsos, C.: PICS: Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs. In: SDM, pp. 439-450. Citeseer, (2012)
11. Ruan, Y., Fuhry, D., Parthasarathy, S.: Efficient community detection in large networks using content and links. In: Proceedings of the 22nd international conference on World Wide Web, pp. 1089-1098. International World Wide Web Conferences Steering Committee, (2013)
12. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. Proceedings of the VLDB Endowment 2, 718-729 (2009)
13. Moser, F., Colak, R., Rafiey, A., Ester, M.: Mining Cohesive Patterns from Graphs with Feature Vectors. In: SDM, pp. 593-604. SIAM, (2009)
14. Fortunato, S.: Community detection in graphs. Physics Reports 486, 75-174 (2010)
15. Schank, T.: Algorithmic aspects of triangle-based network analysis. vol. PhD. Universit¨at Karlsruhe (2007)
16. Chu, S., Cheng, J.: Triangle listing in massive networks and its applications. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. (2011)
17. Wang, N., Zhang, J., Tan, K.-L., Tung, A.K.H.: On triangulation-based dense neighborhood graph discovery. Proceedings of the VLDB Endowment (2010)
18. Klymko, C., Gleich, D., Kolda, T.G.: Using Triangles to Improve Community Detection in Directed Networks. arXiv preprint arXiv:1404.5874 (2014)
19. Prat-Pérez, A., Dominguez-Sal, D., Brunat, J.M., Larriba-Pey, J.-L.: Shaping communities out of triangles. In: Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 1677-1681. ACM, (2012)
20. Serrour, B., Arenas, A., Gómez, S.: Detecting communities of triangles in complex networks using spectral optimization. Computer Communications 34, 629-634 (2011)
21. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. NIPS, pp. 849-856 (2001)
22. Holme, P., Kim, B.J.: Growing scale-free networks with tunable clustering. Physical review E 65, 026107 (2002)
23. Prat-Pérez, A., Dominguez-Sal, D., Larriba-Pey, J.-L.: High quality, scalable and parallel community detection for large real graphs. In: Proceedings of the 23rd international conference on World wide web. (2014)
24. Newman, M.E.: Analysis of weighted networks. Physical Review E 70, 056131 (2004)
25. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the Construction of Internet Portals with Machine Learning. Information Retrieval Journal 3, 127--163 (2000)
26. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the sixth ACM international conference on Web search and data mining, pp. 587-596. ACM, (2013)

Author Profiling: Age Prediction Based on Advanced Bayesian Networks

Seifeddine Mechti [1], Maher Jaoua[2], Rim Faiz[1,3], Heni Bouhamed[2] and Lamia Hadrich Belguith[2]

[1] LARODEC Laboratory, ISG of Tunis B.P.1088, 2000 Le Bardo, Tunisia
[2] ANLP Group, MIRACL Laboratory, University of Sfax, BP 1088, 3018, Sfax
[3]IHEC of Carthage, 2016 Carthage Présidence, Tunisia

mechtiseif@gmail.com,Rim.faiz@ihec.rnu.tn
{maher.jaoua,l.belguith}@fsegs.rnu.tn,
heni_bouhamed@yahoo.fr

Abstract. In this study, we present a new method for profiling the author of an anonymous English text. The aim of author profiling is to determine demographic (age, gender, region, education level) and psychological (personality, mental health) properties of the authors of a text, especially authors of user generated content in social media. To obtain the best classification, authors resort to machine learning methods. Focusing on the works which use the Bayesian networks, all those methods rather apply the Bayesian naïve classifiers which do not yield the best results. Therfore we propose a method based on advanced Bayesian networks for age prediction to over come the mentioned detail problem. We obtained promising results by relying on an English PAN@CLEF 2013 corpus. The obtained results are comparable to the ones obtained by the best state of the art methods. The software and data can be publicly downloaded from www.cicling.org/2016/data/248/CICLING_248.zip.

Keywords: Author profiling, advanced Bayesian networks, age prediction

1 Introduction

There is no doubt that social networks are experiencing significant growth. Social networks require profiling from their users. These users provide false information about themselves. In 2012 Facebook estimate that there were 83 million false profiles[1]. The detection of user profiles in a discussion is an important piece of information for the providers of certain services. This is specifically to study the way in which certain linguistic characteristics vary depending on the profile of the author of a text. Author profiling can be used in other circumstances, for example, in forensic linguistics; the detection of the linguistic profile of the author of a text could be ex-

[1] http://edition.cnn.com/2012/08/02/tech/social-media/facebook-fake-accounts/

tremely valuable for evaluating the suspects. Similarly, in the marketing perspective, companies may be interested in determining what types of people prefer their products. In the literature, many works have focused on the classification of a conversation or a given text and more precisely on the detection of the age, gender, native language and personality of the author [1].

In this paper we present our method for predicting the age of an author based on his/her linguistic attributes. We resorted to the use of advanced Bayesian networks. The paper is organised as follows. Section 2 summarises related research regarding author profiling. Section 3 presents our method of age prediction based on advanced Bayesian networks. Section 4 reports experiments and evaluation carried out using the advanced Bayesians networks. Finally, conclusions are stated and future lines of research are analysed in Section 5.

2 Related work

The detection of the author's profile is the study of how linguistic features vary depending on the authors profiles [2]. The study achieved by the pioneers Koppel et al. has shown that there are linguistic differences between men and women. Indeed, men who prefer to categorize things, use more determiners (the/ this / that, a, etc.) and quantifiers (two, more, a few, etc.). Women, more than men, resort to personal pronouns (I, you, me, etc.) [3]. Argamon [1] worked on the British National Corpus. They used part of speech features. They were able to get 80% accuracy for the prediction of gender. In another study [4] the authors worked on segments of blogs using features such as punctuation, average words/ sentence length, part of speech and word factor analysis. They achieved a gender prediction rate of 72.2%. Peersman [5] used a corpus of Netlog trying unigrams, bigrams, trigrams, and tetagrams. They obtained an average accuracy of 88.88% for the prediction of the authors' age and gender.

In [6], the authors worked on the automatic classification of emails; they got a rate of 81.5% of well classified documents for the gender dimension and 72% for the age dimension. The works of [7]; [8] showed promising results regarding the detection of the author's gender in chats. Recently, [9] tried to perform the prediction of age in conversations among dutch Twitter users. Although the documents are very short (an average length of less than 10 words), 74% of the discussions were highly ranked. In fact, the authors were able to find a mean absolute error between 4.1 and 6.8 years. Pennebaker [10] relies on the change of language features for the prediction of some personality traits of authors in discussions [11]. The author considered unsupervised learning to detect the personality traits of the authors in texts. Besides personality, [12] used the logistic regression method or the binomial model for the detection of the author's native language.

To ensure on effective prediction, authors resort to preprocessing. Indeed, in their work, [13] resort to HTML Cleaning to obtain plain text and discrimination between human-like posts and spam-like posts , while Ashok [14] use the deletion of URLs, hashtags and user entries in Twitter. On the other hand, [15] uses case conversion, invalid characters, multiple white spaces and tokenization and the selection of subcorpus. The study of [3] distinguished two types of attributes: style based features and content-based features. To determine the age or gender of the author of a document, it

is important to consider the function words. Prepositions, pronouns and determiners have shown their effectiveness in an author's profile detection process [16]. In other works, the authors resort to the frequency of punctuation, of capital letters and of citations [17]. HTML attributes such as the URL of an image or the links of a Web page have been used by [18]. In the works of [19], the authors relied on specific vocabulary items (foreign words) to distinguish between authors. These terms are tags in the Stanford Core NLP tagger such as meeee, yessss, thy, u, sisters, etc. Unlike other authors, [20] resort to calculating the frequency of emoticons as one of the discriminating attributes to predict the author's profile. In [21] for instance, the authors resort to Automated Readability Measures such as the readability index, the Coleman-Liau Index, the Rix Readability Index, the Gunning Fog Index and the Flesch-Kinkaid Index. [22] use stylistic features: frequencies of punctuation marks, size of sentences, words that appear once and twice, use of deflections, number of characters, words and sentences. Ashok [14] uses Lexical Analysis such part of speech, proper nouns and character flooding in this choice of attributes and even attributes which are rarely used like those of emoticons have been considered in the work of [2].

In addition to the style used, the content of documents can be of great help in the classification process. What differentiates several age classes, for example, might be the content of their discussions. Indeed [1] distinguished several classes to categorize the authors. For the English language, they identified classes like home, smartphone, games, sports, Job, Marketing, etc. Then, they choose the first k attributes providing the best discrimination. [23] uses content features (n-grams, bag-of-words) while Ashok uses Dictionaries per subcorpus and class, lexical errors, foreign words and specific phrases like : 'my husband', 'my wife',' my son', etc. Finally, [24] uses second order representation based on relationships among terms, documents, profiles and sub-profiles.

However, the major drawback of content based attributes is that they depend on the psychological and mental state of the author (negative emotions, positive emotions) when writing, which might distort the classification results. In order to obtain a prediction for different output classes, several methods and machine learning algorithms were used like Logic Boost, Rotation Forest, Multi-Class Classifier, Multilayer Perceptron, Single Logistic, Logistic Regression, Multinomial Naive Bayes, Random Forest, Support Vector Machines [14]. In another work [22] uses his own frequency-based prediction function. To our knowledge, focusing on the works that use the Bayesian classifiers, we found that all those methods rather apply the Bayesian naive classifiers which do not yield good results for author profiling. In fact, in the work of [23] the author reached 39% accuracy for Blogs, 31% for hotel reviews, and 35% for social media. These results are relatively poor and reflect the ineffectiveness of the naive Bayesian classifiers. As a solution, we resort to advanced Bayesian networks to ameliorate the process of profiling anonymous authors (section 4).

We note that function words serve to express grammatical relationships with other words within a sentence, or specify the attitude or mood of the speaker. Function words might be prepositions, pronouns, auxiliary verbs, conjunctions, grammatical articles or particles, all of which belong to the group of closed-class words.

3 Proposed Method

As shown in Figure 1, our method is composed of four steps:
Preprocessing: The raw text obtained from the crawlers has to be cleaned to re- move noisy data, tags, urls, hashtags etc. The presence of this noisy data could affect and reduce the accuracy of the entire analysis. The cleaned data is then pushed into a database.

Text analysis: We started by calculating the number of occurrences of all words found in the corpus ranking them in order of their appearances. However, we focused onto the first 200 attributes only. We calculated CF (the class frequency) for each class of attributes in order to measure the frequency of occurrence of each class of attributes in each document of the corpus.

Feature set generation: The most common approaches in the literature distinguish two main types of attributes that can be used to detect the author's profile: the stylistic and the content based ones [10]. We manually grouped the terms belonging to the same class of attributes. We identified 15 classes, namely: Prepositions, Pronouns, Determiners, Adverbs, Verbs, No, Of, I, Medicine, Music, Sport, Phone, Beer, Love, Money. For the 'gender' output class, we realized that the purely stylistic attributes yield good results (based on style). Indeed, we selected three attributes: prepositions, pronouns and verbs. These attributes give good performance with decision trees. In addition, for the age output class, we used both of the content based and stylistic based. For this dimension, each age class discusses well-defined topics.

Classification: It is possible to construct an effective classifier using Bayesian networks [25]; [26]; [27]. A Bayesian classifier has n + 1 nodes for a model with n variables. In the classification models, there is necessarily a discrete multinomial central node which has k modalities corresponding to the class; it can be called "class node" and is added to other nodes of descriptive variables. Descriptive variables are denoted by Xi (i from 1 to n). The simple Bayesian classifier structure is that of the naive Bayesian network classifier also called Naive Bayesian (CBN) [28]. For these CBNs, the inter-variable descriptive correlations are not shown and all descriptive variables contribute equally to the classifier. The class node uses the information from each attribute independently of information from other attributes, which is very limited and not optimal for a classification problem. Accordingly, there have been several CBN structure enrichment proposals considering the possible correlations between the descriptive nodes. In [26], the authors proposed the Tree Augmented Naive Bayes method (TAN) in order to enrich the network structure using the shaft structure [29]. [30]. The construction of this structure is not greedy in computational complexity, but the restriction of the number of parents of a node to 2 (1 + class node) represents a real gap and risks taking the model away from reality. The resulting structure represents neither cases where a variable is correlated with several other descriptive variables nor the case where a variable is independent of all the others (in this case the node representing this variable only needs the class node as parent and the addition of another parent node only increases the complexity of the learning of the settings). Consequently, other authors proposed the use of the Augmented Naive Bayesian (BAN)

networks [26]; [31] where the addition of arcs between descriptive variable nodes is carried out with algorithms which do not impose any restrictions. Other authors simply proposed the use of general methods for the learning of Bayesian network structures (GBN) [26]; [31]) where the class node is regarded as an ordinary node and is not automatically connected with all other nodes [32]; [30]; [33]; [34]; [35]; [36]. It is true that thanks to the last we obtain closer to reality Bayesian classifier structures and therefore, the possibility of have more efficient classifiers. Hence, for the building and operation of the Bayesian network, we will use the Bayesian network toolbox (BNT) [37] running with the matlab software (Version 2010). Specifically, we will use the "Greedy Search" (GS) [38] for the learning structure and the 'Click-tree propagation algorithm [39] for the inference. We used a portion of 30000 examples for the learning phase and another portion of 30000 examples for the test phase of the classifier.

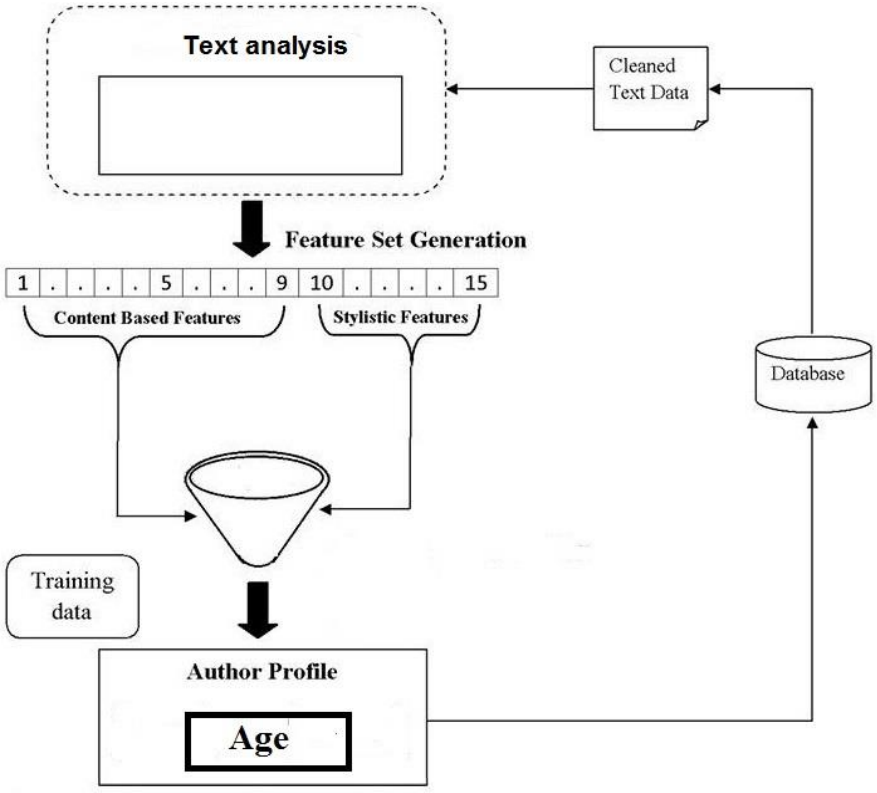


Fig. 1. System architecture diagram

4 Experimentation and Evaluation

4.1 Dataset Description

In our data, the adopted documents are blog posts written in English. The variety of themes provides a wide spectrum of topics, making the task of determining age and

gender more realistic. The age groups were defined according to [40]: the 10s is the class of individuals between 13 and 17 years old, the 20s are those between 20 and 33 year olds, and finally the 30s are those between 33 and 47 years old. Table 1 summarizes the contents of the corpus. We note that each file has a different author and more files cannot have the same author.

The corpus consists of 236600 files for training and 25440 for testing. For machine-learning, the class of 30s includes 133508 authors unlike the class of 10s which includes only 17200 files. The corpus is balanced in terms of gender but imbalanced in terms of age.

Table 1. Dataset description

		Number of authors	
Age	Gender	Training	Test
10S	Male	8600	888
	Female	8600	888
20S	Male	42828	4576
	Female	42875	4598
30S	Male	66708	7184
	Female	66800	7224
Total		236600	25440

4.2 Baseline Method

For comparison purposes a baseline was used so as to evaluate one's own results. We rely on the results of [41] in PAN@CLEF2013[2] as a baseline method. They ranked 3rd in this competitive conference. Using the free learning software Weka[3], this method started with the construction of ARFF (Attribute Relationship File Format) age dimension. The features are collected and then fed into an ensemble classifier. For categorization, authors used decision trees classifier (J48) due to speed and accuracy. The classifier is trained with the whole data corpus and used later for testing purposes. They got a good classification rate of 0.58.

4.3 Results Based on Advanced Bayesian Networks

Based on the advanced Bayesian networks, the proposed method has good performance. According to the confusion matrix, for the age prediction we got a good classification rate of 0,6175. Compared to the results reached with the decision trees, we notice the added values brought about by the Bayesian networks in this classification. Also, a good classifier is expected to yield the best recall measure. Indeed, the classifier retrieves 74,5% of the relevant documents against 55% with decision trees.

[2] http://pan.webis.de/clef13/pan13-web/author-profiling.html

Table 2. Confusion matrix for age prediction

	10s	**20s**	**30s**	**Total**	**Accuracy**
1	3124	1782	5094	10000	0.3124
2	0	5537	4463	10000	0.5537
3	0	132	9868	10000	0.9868
Total	3124	7451	19425	30000	**0.6176**

5 Conclusion

In this study, we have performed a document categorization so as to provide an author profile classification according to his/her text's characteristics. Content based attributes could be discriminative elements in the documents partitioning among age classes. Such a deduction can be predicted since children, the middle aged adults and elderly people never discuss the same topics. The improvements of our performance are mainly due to the proposition of a new method based on advanced Bayesian networks for classification. The performances of these networks prove their effectiveness in terms of accuracy and recall. It can be concluded that the use of the lexical classes is not enough. That is why, and as a perspective, we intend to integrate other aspects like syntax, morphology and semantics. Furthermore, to allow a better author detection we think of going beyond the age dimension and consider the detection of the native language and geographical data of the author and above all the detection of his/her personality. The software and data can be publicly downloaded from www.cicling.org/2016/data/248/CICLING_248.zip.

References

1. Argamon, S., Koppel M., Pennebaker J., and Schler J. Automatically detection the author of an anonymous text. Communications of the ACM, p 119-123. 2009.
2. Maharjan S., Shrestha P, and Solorio T. A Simple Approach to Author Profiling in MapReduce. England. CLEF 2014.
3. Koppel M. Argamon S. and Shimoni A., Automatically categorizing written texts by author, gender, Literary and Linguistic Computing, pages 401-412, 2003.
4. Zhang., C and Zhang.,P .Predicting gender from blog posts. Technical report, Technical Report. University of Massachusetts Amherst, USA, 2010.
5. Peersman C., Daelemans., W, and Van Vaerenbergh., L. Predicting age and gender in online social networks. In Proceedings of the 3rd international workshop on Search and mining user-generated contents, SMUC 11, pages 37-44, New York, NY, USA, ACM.2011.
6. Gaustad T., Estival D. and Hutchinson B. TAT: an author profiling tool with application to Arabic emails. Proceedings of the Australsian Language Technology Workshop, pages 21-30, Melbourne, Australia, 2007.
7. Hariharan, S.,Gender Prediction in Chat based Medium s Using Text Mining, in: International Journal of Research and Reviews in Information Sciences (IJRRIS), Pakistan, 2011.
8. Kose C., Ozyurt O, and Amanmyradov G. Mining Chat Conversations for Sex Identification, Emerging Technologies in Knowledge Discovery and Data Mining (PAKKD), Nanjing, China, 2007.

9. Nguyen D, Gravel R, Trieschnigg D, and Meder T. "how old do you think i am?"; a study of language and age in twitter. Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, 2013.
10. Pennebaker., J. The secret life of pronouns: What our words say about us. pp. 401-412. 2011.
11. Celi F. Unsupervised Personality Recognition from Text: Possible Applications. In proceeding of PAN at CLEF, England. 2014.
12. Sze-Meng Jojo., W and Dras.,M. Contrastive analysis and native language identification. In Proceedings of the Australasian Language Technology Association Workshop, pages 53-61, Sydney, Australia. 2009.
13. Marquardt., J, Fanardi,G., Vasudevan, G., Moens., M, Davalos., S, Teredesai, D and De Cock. M. Age and Gender Identification in Social Media.PAN at CLEF 2014.
14. Rangel F, Rosso P, Koppel M, Stamatatos E, and Inches G. Overview of the Author Profiling Task. PAN@CLEF. Valencia, spain .2013.
15. Edson R. D. Weren, Viviane P. Moreira, and Jose P. M. de Oliveira. Using Simple Content Features for Author Profiling PAN@CLEF.Valencia,Spain.2013.
16. Fermin, L., Cruz., Rafa Haro R., and Javier Ortega, F. ITALICA at PAN 2013: An Ensemble Learning Approach to Author Profiling. Notebook for PAN at CLEF .Valencia. 2013.
17. Aleman Y., Loya N, Vilarino D, and Pinto D. Two methodologies applied to the author profiling task. Notebook for PAN at CLEF. Spain. 2014.
18. Sapkota ,U., Solorio, T., Montes-y-Gomez, M, and De-la-Rosa, G. Author Profiling for English and Spanish Text .PAN at CLEF. Spain. 2013.
19. Gopal Patra, B., Banerjee, S, Das., D, Tanik., S, and Sivaji Bandy.,O . Automatic Author Profiling Based on Linguistic and Stylistic Features. PAN at CLEF 2013.
20. Irazu D., Farias H., Guzman-Cabrera R.,Reyes A and Rocha M. Semantic-based Features for Author Profiling Identification: First insights- Notebook for PAN at CLEF. Spain. 2013.
21. Gressel., G, Hrudya P, Surendran K, Thara S, Aravind A, and Prabaharan P. Ensemble Learning Approach for Author Profiling.PAN at CLEF.England. 2014.
22. Baker ., C. Proof of Concept Framework for Prediction.Pan@CLEF. England. 2014.
23. Villena-Roman,J and Gonzalez-Cristobal,J. DAEDALUS: Guessing Tweet Author's Gender and Age. PAN@CLEF.England.2014.
24. Pastor L, Montes-Y-Gomez M, Escalante H, Villasenor-Pineda L and Villatoro-Tello E. INAOE's Participation at PAN'13: Author Profiling task—Notebook for PAN at CLEF 2013.
25. Langley, P., Selection of relevant features in machine learning. Acts of AAAI Fall Symposium on Relevance, p. 140-144.1994.
26. Friedman, N., Geiger, D., Goldszmid, M., Bayesian Network classifiers, Machine Learning, p. 131-163.1997.
27. Pernkopf, F., Bayesian network classifiers versus selective k-NN classifier, Pattern Recognition, p. 1-10.2005.
28. Domingos, P., Pazzani, M., On the optimality of the simple Bayesian classifier under zero-one loss, MachineLearning, 1997, p. 103-130.
29. Chow, C., Liu, C., Approximating discrete probability distributions with dependence trees, IEEE Transactions on Information Theory, 14 (3), p. 462-467.1968.
30. Madden, M. G., A New Bayesian Network Structure for Classification Tasks , Actes de la 13th Irish Conference on Artificial Intelligence and Cognitive Science, 2002, p.203-208.
31. Cheng, J., Greiner, R., Learning Bayesian belief network classifiers: algorithms and system , Actes de la 14th Canadian Conference on Artificial Intelligence, 2001, p. 141-151.
32. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P.,Optimization by simulated annealing , Science, , 1983, p. 671-681.

33. Stuart, M., Yulan, H., Kecheng, L.,Choosing the best Bayesian classifier : An empirical study, IAENG International Journal of Computer Science, 2009, p. 1-10.
34. Carta, J. A., Velazquez, S., Matias, J. M., Use of Bayesian networks classifiers for long term mean wind turbine energy output estimation at a potential wind energy conversion site , Energy Conversion and Management,p.1137-1149. 2011.
35. Lerner, B., Malka, R., Investigation of the K2 algorithm in learning Bayesian Network Classifiers, Applied Artificial Intelligence, p.74-96.2011.
36. Bouhamed, H., Masmoudi, A., Lecroq, T., Rebai A., Reducing the structure space of Bayesian classifiers using some general algorithms, Journal of Mathematical Modelling and Algorithms in Operations Research(Springer), Volume 14, Issue 2, pages 197-237.2015.
37. Murphy, K, The BayesNet Toolbox for Matlab , Computing Science and Statistics : Interface .33,http://www.ai.mit.edu/murphyk/Software/BNT/bnt.html.2001.
38. Chickering, D., A Transformational Characterization of Equivalent Bayesian Network Structures, Actes de la 11th Conference on Uncertainty in Artificial Intelligence (UAI'95), San Francisco, CA, USA, Morgan Kaufmann Publishers, , p. 87-98.1995.
39. Lauritzen, S., Spiegelhalter, D., Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, 50, p. 157-224.1988.
40. Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. Effects of age and gender on blogging. AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs, 2006.
41. Mechti S., Jaoua M. Belguith L and Faiz R., Author profiling using style based features. In proceeding of PAN at CLEF, Spain. 2013.

The Key Factors and Their Influence in Authorship Attribution

Raheem Sarwar and Sarana Nutanong

City University of Hong Kong
Department of Computer Science
Hong Kong
rsarwar2-c@my.cityu.edu.hk, snutanong@cityu.edu.hk

Abstract. Authorship attribution has a long history started since 19th century. Existing studies have used different sets of stylometric features and computational methodologies on a variety of corpus with different lengths and genres. This study presents a protocol to perform a *systematic literature review (SLR)* to identify the best combination of stylometric features and computational methodology. Specifically, we formulate an SLR protocol that can be used to conduct a literature survey to help answer like (i) whether it is possible to identify the authorial style of an author regardless the genre and length of the text, and (ii) how to select specific stylometric features and computational methodology. We also conduct an example of how the proposed SLR protocol can be used as a template for publication extraction and filtering for an SLR on authorship attribution.
Keywords: Authorship attribution, Stylometric features, Computational methodologies

1 Introduction

Authorship Attribution (AA) problem is generally expressed as: given a disputed text and a set of candidate authors with their writing samples, find the author of the given disputed text from the set of candidate authors [1]. AA has a very long history started from 19th century and many approaches have been proposed for it. Existing approaches can be divided into two main tasks. Finding appropriate features of the language to quantify the writing style of authors, and forming efficient approaches to apply these features. A lot of stylometric features have been proposed so far including word lengths, sentence lengths, vocabulary richness and character frequencies. Rudman (1998) reported that almost 1 thousand measures has been proposed to quantify the writing styles of the authors [2]. During the last decade, this research areas has been extensively investigated by researchers in the fields of natural language processing [3,4], machine learning [5] and information retrieval [6].

Existing studies of AA used different sets of stylometric features and computational methodologies on a variety of corpus with different length and genre of the text [1,2,4,7–30]. We have formulated following research questions to address in the systematic literature review (SLR):

- **RQ1:** Which combination of the "set of stylometric features" and computational methodology is best in terms of accuracy in AA, and reasoning?
- **RQ2:** Is it possible in AA to identify the authorial style of the author regardless of the genre and length of corpus and without selecting specific stylometric features and computational methodology, and reasoning?

This paper presents a Systematic Literature Review (SLR) protocol to address key research questions in Authorship Identification. SLR is used for identification, evaluation and interpretation of all available research to specific research questions. To the best of our knowledge, this is the first SLR protocol formulated to address the authorship identification problem. Note that the nature of contribution of our work is introducing the SLR protocol rather than the study initiated from the protocol. The nature of our investigation is similar to the SLR protocols proposed by these publications [31–35]. Writing SLR protocol is important before we start the detailed review because the thoroughness of the protocol will ensure that the process remains rigorous. Developing an SLR protocol is consider prerequisite for detailed literature review in an area. A detailed review will be conducted as future work. The resultant protocol obtained from this investigation can be used to help investigate the scope of primary studies in which empirical evidence "contradicts" or "supports" with our theoretical hypotheses and to help generate new hypotheses. Specifically, our SLR protocol provides (i) a systematic means to select related studies in order to reduce biases through a well defined and comprehensive methodology; (ii) the information about the influence of some phenomenon based on empirical methods and wide range of settings. A consistent SLR study also provides evidence that phenomenon is transferable and robust, otherwise, the sources of the variations can be explored [36]. The rest of the paper is organized as follows. Section 2 presents the literature review. Section 3 presents the formation of the SLR protocol to answer the proposed research questions. Section 4 presents the preliminary results of this study. Section 5 presents the conclusion and future work.

2 Literature Review

Authorship attribution has a very long history started since 19th century. The first attempt to identify the author based on the writing style was made by Mendenhall [37] in 1887 followed by Zipf [8] and Yule [7] in 1932 and 1939, respectively. Later on, this problem was solved by performing the Bayesian statistical analysis on the frequencies of common words e.g., 'to', 'and' etc by Mosteller and Wallace [9] in 1964. Subsequently, Holmes [38] formulated a feature set to quantify the writing styles of the authors which is also known as Stylometry. The study of stylometry is concerned with statistical analyses of variations in the author's literary style (represented as a set of features), which remains relatively unchanged across different documents [10, 38]. Thus far, a variety of stylometric features have been proposed for AA including average sentence length, average word length [13], vocabulary richness [14], frequencies of punctuation [13], word

endings [15], character n-grams [39], word n-grams [40], parts of speech n-grams [15], the organization of words, vocabulary distributions and the number of occurrences of particular word [41]. During the last decade, this research areas has been extensively investigated by researchers in the fields of natural language processing [3,4], machine learning [5] and information retrieval [6]. There are many techniques from machine learning and artificial intelligence that have been used for AA. In earlier days, the Bayesian statistical analysis [9] was used for authorship attribution; the recent techniques which have been used for authorship attribution include support vector machines [39] neural network [13, 42], radial basis function networks [19], decision trees [18], and nearest neighbor classification [1]. Moreover, the markov chains [43], principal component analysis [17] and compression based techniques have also been used for AA [16].

3 Systematic Literature Review Protocol

According to Kitchenham [36], a systematic literature review (SLR) have three steps: (i) planning a review, (ii) conducting the review, and (iii) reporting the review. This paper focuses only on the first step, planning of a review, i.e., formulating an SLR protocol to address the research questions with preliminary results. An SLR protocol explains the methodology to conduct a literature review. The protocol decrease researchers bias to a specific set of publications [31]. For instance, without a predefined protocol, there is a possibility that the selection of primary studies may be driven by the expectations of the researcher [44]. Figure 1 shows the steps of the SLR protocol. The first and most important step is the formation of research questions. The next step is concerned with defining the search strategy (research process) to retrieve the primary research studies by exploring different publisher sites and index engines. The third step provides a method of how to filter irrelevant and less important studies. The next step involves assessing the quality of the selected primary research studies. Finally, the data collection and synthesis are performed. The details of each step is discussed in the following sections.

Fig. 1. Development of the Systematic Literature Review Protocol

3.1 Search Strategy

As explained in Section 1, we consider two research questions, RQ1 and RQ2.

We have developed the following strategy to formulate search queries to retrieve the primary studies to conduct the review.

1. **Derive Keywords:** Derive the main keywords from each research question.
2. **Derive Alternative Words:** Derive the alternative words or synonyms for each keyword obtained from research questions.
3. **Verification of Keywords:** Verify each keyword from the literature to ensure their correctness.
4. **Use Boolean Operators:** If bibliographic database provide the option, use Boolean "OR" operator to integrate alternative keywords and synonyms, and use the "AND" operator to integrate the major terms.

3.1.1 Results for 1 (Derive Keywords)

- **RQ1:** Stylometric Features, Computational Methodology, Authorship Attribution, Accuracy.
- **RQ2:** Authorship Attribution, Authorial Style, Author, Genre, Length, Corpus, Stylometric Features, Computational Methodology.

3.1.2 Results for 2 (Derive Alternative Keywords)

- **RQ1:**
- **Stylometric Features:**
- ("Stylometric Features" OR "authorial features" OR "stylometric properties" OR "stylometric analysis" OR "stylometric identification" OR "stylistic fingerprints" OR "linguistic fingerprint" OR "linguistic features ")
- **Computational Methodology:**
- ("Computational Methodology" OR "machine learning" OR "information retrieval" OR "bayesian statistical analysis" OR "support vector machines" OR "neural network" OR "radial basis function networks" OR "decision trees" OR "nearest neighbor classification" OR "markov chains" OR "principal component analysis" OR "compression based techniques" OR "latent dirichlet allocation" OR "feature transformation" OR "feature selection" OR "clustering" OR "supervised learning" OR "unsupervised learning" OR "semi supervised ensemble algorithm" OR "deep learning algorithm" OR "association rule" OR "instance based" OR "natural language processing" OR "statistical analysis")
- **Authorship Attribution:**
- ("Authorship Attribution" OR "author identification" OR "author recognition" OR "disputed authorship" OR "forensic authorship analysis" OR "author identity resolution" OR "stylometric identification")
- **Accuracy:**
- ("Accuracy" OR "enhance" OR "effective" OR "scalable" OR "experiment" OR "precision" OR "recall" OR "accurateness"
- **RQ2:**
- **Authorship Attribution:**

– ("Authorship Attribution" OR "author identification" OR "author recognition" OR "disputed authorship" OR "forensic authorship analysis" OR "author identity resolution" OR "stylometric identification")
– **Authorial Style:**
– ("Authorial Style" OR "literary style")
– **Author:**
– ("Author" OR "writer" OR "novelist" OR "biographer" OR "essayist" OR "dramatist" OR "playwright")
– **Genre:**
– ("Genre" OR "type" OR "kind" OR "field" OR "email" OR "plays" OR "formal" OR "informal" OR "social Media")
– **Length:**
– ("Length" OR "size" OR "short" OR "long" OR "chunk")
– **Corpus:**
– ("Corpus" OR "text" OR "resource" OR "data")
– **Stylometric Features:**
– ("Stylometric Features" OR "authorial features" OR "stylometric properties" OR "stylometric analysis" OR "stylometric identification" OR "stylistic fingerprints" OR "linguistic fingerprint" OR "linguistic features ")
– **Computational Methodology:**
– ("Computational Methodology" OR "machine learning" OR "information retrieval" OR "bayesian statistical analysis" OR "support vector machines" OR "neural network" OR "radial basis function networks" OR "decision trees" OR "nearest neighbor classification" OR "markov chains" OR "principal component analysis" OR "compression based techniques" OR "latent dirichlet allocation" OR "feature transformation" OR "feature selection" OR "clustering" OR "supervised learning" OR "unsupervised learning" OR "semi supervised ensemble algorithm" OR "deep learning algorithm" OR "association rule" OR "instance based" OR "natural language processing" OR "statistical analysis")

3.1.3 Results for 3 (Verification of Keywords)

– The correctness of all keywords from research questions have been verified from existing studies of Authorship Attribution.

3.1.4 Results for 4 (Use Boolean Operators)

– **RQ1:** ("Stylometric Features" OR "authorial features" OR "stylometric properties" OR "stylometric analysis" OR "stylometric identification" OR "stylistic fingerprints" OR "linguistic fingerprint" OR "linguistic features ") AND ("Computational Methodology" OR "machine learning" OR "information retrieval" OR "bayesian statistical analysis" OR "support vector machines" OR "neural network" OR "radial basis function networks" OR "decision trees" OR "nearest neighbor classification" OR "markov chains" OR "principal component analysis" OR "compression based techniques"

OR “latent dirichlet allocation” OR “feature transformation” OR “feature selection” OR “clustering” OR “supervised learning” OR “unsupervised learning” OR “semi supervised ensemble algorithm” OR “deep learning algorithm” OR “association rule” OR “instance based” OR “natural language processing” OR “statistical analysis”) AND (“Authorship Attribution” OR “author identification” OR “authorship analysis” OR “author recognition” OR “disputed authorship” OR “authorship verification” OR “intrinsic plagiarism” OR “unconscious authorship” OR “obfuscate authorship” OR “marker of authorship” OR “analysis of authorship” OR “computational analysis of authorship” OR “linguistic pattern recognition” OR “forensic authorship analysis” OR “fighting authorship” OR “author identity resolution” OR “author profiling” OR “stylometric identification”) AND (“Accuracy” OR “enhance” OR “effective” OR “scalable” OR “experiment” OR “effect” OR “precision” OR “accurateness” OR “optimization” OR “robustness”) AND (“Reasoning” OR “causes” OR “basis” OR “root” OR “origin” OR “source”)

– **RQ2:** (“Authorship Attribution” OR “author identification” OR “authorship analysis” OR “author recognition” OR “disputed authorship” OR “authorship verification” OR “intrinsic plagiarism” OR “unconscious authorship” OR “obfuscate authorship” OR “marker of authorship” OR “analysis of authorship” OR “computational analysis of authorship” OR “linguistic pattern recognition” OR “forensic authorship analysis” OR “fighting authorship” OR “author identity resolution” OR “author profiling” OR “stylometric identification”) AND (“Authorial Style” OR “literary style” OR “authorial component”) AND (“Author” OR “writer” OR “novelist” OR “biographer” OR “essayist” OR “dramatist” OR “playwright”) AND (“Genre” OR “type” OR “kind” OR “field” OR “email” OR “plays” OR “formal” OR “informal” OR “social media” OR “contemporary American English”) AND (“Length” OR “size” OR “short” OR “long” OR “huge” OR “chunk”) AND (“Corpus” OR “text” OR “resource” OR “data”) AND (“Stylometric Features” OR “authorial features” OR “stylometric properties” OR “stylometric analysis” OR “stylometric identification” OR “stylistic fingerprints” OR “linguistic fingerprint” OR “linguistic features ”) AND (“Computational Methodology” OR “machine learning” OR “information retrieval” OR “bayesian statistical analysis” OR “support vector machines” OR “neural network” OR “radial basis function networks” OR “decision trees” OR “nearest neighbor classification” OR “markov chains” OR “principal component analysis” OR “compression based techniques” OR “latent dirichlet allocation” OR “feature transformation” OR “feature selection” OR “clustering” OR “supervised learning” OR “unsupervised learning” OR “semi supervised ensemble algorithm” OR “deep learning algorithm” OR “association rule” OR “instance based” OR “natural language processing” OR “statistical analysis”) AND (“Reasoning” OR “causes” OR “basis” OR “root” OR “origin” OR “source”)

3.2 Resources to be Searched

Different bibliographic databases are selected to extract relevant conference papers and journal articles. Bibliographic databases are chosen on the basis of research experience, preferences or suggested by other researchers and personal knowledge [45].

The resources utilized in this study are shown in Table 1.

Table 1. Resources to be searched

Publisher's Site	Index Engines
ACM Digital Library	Scopus
IEEE Xplore	Compendex
Wiley Inter Science	Google Scholar
Science Direct	Cite Seer
Springer Link	Inspec
Business Source Premier	ISI Web of Science

3.3 Documentation of Search Results

The documentation of the search results is important to make the query process precise and replicable [45]. During the systematic literature review, the following data of the retrieved publications will be recorded: Serial No, Bibliographic Database, Query Date, Search Strategy, Search String, Years, Number of publications retrieved, Initial Selection Decision, Final Selection Decision.

3.4 Publication Selection Criteria

Publication selection criteria is used to decide which research papers are included in, or excluded from, a systematic literature review. It helps to pilot the selection criteria for review on a subset of primary publications

3.4.1 Inclusion Criteria: The inclusion criteria used in this paper helps to determine which research paper should be considered for review. In this study only those articles, reports and research papers will be considered in which stylometric features are used for Authorship Attribution on the text of different genre and length. The inclusion criteria is as follows:

- Studies that use stylometric features Authorship Attribution.
- Studies that clearly describe the reasons of selecting a particular set of stylometric features and computational methodology.
- Studies that perform Authorship Attribution on the corpus of different length and genre.
- Studies that clearly describe the affect of length and genre of the text on the accuracy.

3.4.2 Exclusion Criteria: The following exclusion criteria is used to eliminate the irrelevant literature from selected research papers:

- Studies that does not focus on authorship attribution.
- Studies that are not written in English.
- Research work that does not highlight the affect of genre and length of the text and the selection of particular set of stylometric features and computational methodologies for Authorship Attribution.
- Primary literature will be reviewed on the basis of criteria mentioned in Table 2. The existing individual research papers contributing to a SLR is named as primary research; a SLR is the form of secondary study.

Table 2. Review Procedure of Primary Studies

Relevance Analysis Phase.	**Inclusion and Exclusion Criteria**
Uniqueness	Ensure the uniqueness of the publication. They must be written in English
Relevance	Read the title and abstract to ensure the relevance of the study with our research question, in case of ambiguity, go through introduction and conclusion of the publication
Full Text	Select the studies after reading full text

3.4.3 Publication Quality Assessment: The publication quality assessment (PQA) of selected papers will take place after applying relevance and selection criteria mentioned in Table 2. The PQA of the selected publications will be performed parallel to the phase of data extraction. For PQA the following research questions have been taken under consideration:

- Does the paper clearly describe the stylometric features and computational methodology adopted to perform Authorship Attribution as there are some studies which do not list the stylometric features adopted to conduct the study.
- Does the research paper clearly describe the reason to select the specific set of stylometric feature for a specific kind and length of text.
- Does the study compare the result with existing techniques.
- Is the researcher seems biased to mention positive results more than negative results?

  Each of the above point will be graded as “No” or “Yes” or “partial” or “N.A”.

3.4.4 Data Extraction: Data extraction is concerned with defining a procedure to get the relevant data from selected primary studies. A data extraction

form is used to collect data from the selected studies to perform systematic literature review. Before the phase of data extraction, we will implement pilot data extraction. The review of selected primary studies will be undertaken by a single researcher who is responsible for data extraction. In case of an issue concerning the data extraction, a secondary reviewer will be approached for the guidance.

3.5 Data Synthesis

Data synthesis involves collecting and summarising the results of selected primary studies. The synthesis of extracted data can be categorized into five parts. The first part consists of stylometric features. The second part consists of the computational methodologies performed on these stylometric features. The third part is concerned with the effect of genre on stylometric features and computational methodologies. The fourth part is concerned with the effect of the length of the text on stylometric features and computational methodologies. The final part provides quantitative analyses on the results. The data for these five parts are synthesized and presented in the format similar to that of Table 3. In Table 3, the Frequency is the ratio of the primary studies which presents search area and the total number of selected primary studies. The percentage represents the percentage of the total primary studies in which the required information is clearly described.

Table 3. Data synthesis format. This table is only an example of format and intentionally does not present any specific figures.

Search Area	**Paper Title**	**Authors**	**Years**	**Frequency**	**Percentage**
Stylometric Features	Title 1	Authors	...	Freq. 1	...
	Title 1	Authors	...	Freq. 1	...
	.	.		.	
	.	.		.	
	.	.		.	
	Title n	Author n		Freq. n	
Computational Methodologies	...	...	...	...	...
Genre	...	...	...	...	...
Length	...	...	...	...	...
Reasoning	...	...	...	...	...

4 Preliminary Results

We are currently in the implementation phase of the SLR and we have got results for some of the aforementioned sections of the proposed protocol. These are Sections 3.2, 3.3 and 3.5. After applying the aforementioned search strategy mentioned in Section 3.2 on the specified bibliographic resources, we selected

some primary studies from retrieved studies based on the inclusion/exclusion criteria mentioned in 3.5. The preliminary results of this study based on existing studies can be summarised as follows:

- **RQ1:**The selection of the stylometric features affects the accuracy of the authorship attribution (AA). Moreover, the selection of appropriate computational methodology for a specific set of stylometric feature increases the accuracy of the AA [1, 2, 7–19].
- **RQ2:** The text of different genres require different set of stylometric features to obtain satisfactory results [4, 20, 21]. Genre-dependent stylometric features outperform the genre-independent stylometric features [22, 27, 30]. The accuracy of AA task is highly dependent on the length of the text, long text produce satisfactory results as compared to short text with same set of stylometric features, however, satisfactory results can be obtained with short text by selecting appropriate stylometric features [7, 26].

The preliminary results mentioned above will help to answer our proposed research question in the SRL.

5 Conclusions and Future Work

In this study, we propose a systematic literature review (SLR) protocol to identify the key factors and their influence in the field of authorship attribution. Our proposed SLR protocol can be used to create a well-defined literature survey in the area of Authorship Attribution. Specifically, we focus on exploring different stylometric feature sets and computational methodologies that can be adopted to increase the accuracy of Authorship Attribution. Our protocol can be used to define the scope of primary studies in which empirical evidence "contradicts" with or "supports" our theoretical hypotheses and will help to generate new hypotheses. As future work, we plan to apply our proposed protocol to conduct a comprehensive SLR study on our proposed research questions.

References

1. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. Journal of the American Society for information Science and Technology **60** (2009) 9–26
2. Rudman, J.: The state of authorship attribution studies: Some problems and solutions. Computers and the Humanities **31** (1997) 351–365
3. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Computer-based authorship attribution without lexical measures. Computers and the Humanities **35** (2001) 193–214
4. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic text categorization in terms of genre and author. Computational linguistics **26** (2000) 471–495
5. Khosmood, F., Levinson, R.: Toward unification of source attribution processes and techniques. In: Machine Learning and Cybernetics, 2006 International Conference on, IEEE (2006) 4551–4556

6. Zhao, Y., Zobel, J.: Effective and scalable authorship attribution using function words. In: Information Retrieval Technology. Springer (2005) 174–189
7. Yule, G.U.: On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. Biometrika **30** (1939) 363–390
8. Zipf, G.K.: Selected studies of the principle of relative frequency in language. (1932)
9. Mosteller, F., Wallace, D.: Inference and disputed authorship: The federalist. (1964)
10. Holmes, D.I.: The evolution of stylometry in humanities scholarship. Literary and linguistic computing **13** (1998) 111–117
11. Argamon, S., Levitan, S.: Measuring the usefulness of function words for authorship attribution. In: ACH/ALLC. (2005)
12. Argamon, S.: Interpreting burrows's delta: geometric and probabilistic foundations. Literary and Linguistic Computing **23** (2008) 131–147
13. Graham, N., Hirst, G., Marthi, B.: Segmenting documents by stylistic character. Natural Language Engineering **11** (2005) 397–415
14. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proceedings of the twenty-first international conference on Machine learning, ACM (2004) 62
15. Madigan, D., Genkin, A., Lewis, D.D., Argamon, S., Fradkin, D., Ye, L.: Author identification on the large scale. In: Proc. of the Meeting of the Classification Society of North America. (2005) 13
16. Benedetto, D., Caglioti, E., Loreto, V.: Language trees and zipping. Physical Review Letters **88** (2002) 048702
17. Juola, P., Baayen, R.H.: A controlled-corpus experiment in authorship identification by cross-entropy. Literary and Linguistic Computing **20** (2005) 59–67
18. Uzuner, Ö., Katz, B.: A comparative study of language models for book and author recognition. In: Natural Language Processing–IJCNLP 2005. Springer (2005) 969–980
19. Pandian, A., Sadiq, M.A.K.: Authorship categorization in email investigations using fisher's linear discriminant method with radial basis function. Journal of Computer Science **10** (2014) 1003
20. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Text genre detection using common word frequencies. In: Proceedings of the 18th conference on Computational linguistics-Volume 2, Association for Computational Linguistics (2000) 808–814
21. Argamon, S., Koppel, M., Fine, J., Shimoni, A.R.: Gender, genre, and writing style in formal written texts. TEXT-THE HAGUE THEN AMSTERDAM THEN BERLIN- **23** (2003) 321–346
22. Amasyalı, M.F., Diri, B.: Automatic turkish text categorization in terms of author, genre and gender. In: Natural Language Processing and Information Systems. Springer (2006) 221–226
23. Luyckx, K., Daelemans, W.: The effect of author set size and data size in authorship attribution. Literary and linguistic Computing (2010) fqq013
24. Hirst, G., Feiguina, O.: Bigrams of syntactic labels for authorship discrimination of short texts. Literary and Linguistic Computing **22** (2007) 405–417
25. Mikros, G.K., Argiri, E.K.: Investigating topic influence in authorship attribution. In: PAN. (2007)
26. Eder, M.: Does size matter? authorship attribution, small samples, big problem. Proceedings of Digital Humanities (2010) 132–135
27. Rybicki, J., Eder, M.: Deeper delta across genres and languages: do we really need the most frequent words? Literary and Linguistic Computing (2011) fqr031

28. Jockers, M.L., Witten, D.M.: A comparative study of machine learning methods for authorship attribution. Literary and Linguistic Computing (2010) fqq001
29. Houvardas, J., Stamatatos, E.: N-gram feature selection for authorship identification. In: Artificial Intelligence: Methodology, Systems, and Applications. Springer (2006) 77–86
30. Kelih, E., Antić, G., Grzybek, P., Stadlober, E.: Classification of author and/or genre? the impact of word length. In: Classification—the Ubiquitous Challenge. Springer (2005) 498–505
31. Khan, A., Basri, S., Amin, F., Teknologi, U., Perak, T., Studies, I.: Communication risks and best practices in global software development during requirements change management: A systematic literature review protocol. Research Journal of Applied Sciences, Engineering and Technology **6** (2013) 3514
32. Rehman, S., Khan, S.U.: Swot analysis of software quality metrics for global software development: A systematic literature review protocol. IOSR Journal of Computer Engineering **2** (2012)
33. Khan, S.U., Niazi, M., Ikram, N.: Software development outsourcing relationships trust: a systematic literature review protocol. Evaluation and Assessment in Software Engineering, EASE (2010)
34. Alam, A.U., Khan, S.U.: Knowledge sharing management in offshore software development outsourcing relationships from vendors' perspective: A systematic literature review protocol. In: Software Engineering (MySEC), 2011 5th Malaysian Conference in, IEEE (2011) 469–474
35. Qureshi, N., Ikram, N., Bano, M., Usman, M.: Empirical evidence in software architecture: a systematic literature review protocol. In: The Sixth International Conference on Software Engineering Advances. (2011) 534–538
36. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33** (2004) 1–26
37. Mendenhall, T.C.: The characteristic curves of composition. Science (1887) 237–249
38. Holmes, D.I.: Authorship attribution. Computers and the Humanities **28** (1994) 87–106
39. Hedegaard, S., Simonsen, J.G.: Lost in translation: Authorship attribution using frame semantics. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, Association for Computational Linguistics (2011) 65–70
40. Escalante, H.J., Solorio, T., Montes-y Gómez, M.: Local histograms of character n-grams for authorship attribution. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics (2011) 288–298
41. Juola, P.: Authorship attribution for electronic documents. In: Advances in digital forensics II. Springer (2006) 119–130
42. Zheng, R., Li, J., Chen, H., Huang, Z.: A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology **57** (2006) 378–393
43. Baayen, H., van Halteren, H., Neijt, A., Tweedie, F.: An experiment in authorship attribution. In: 6th JADT. (2002) 29–37
44. Khan, A.A., Basri, S., Dominic, P.: Communication risks in gsd during rcm: Results from slr. In: Computer and Information Sciences (ICCOINS), 2014 International Conference on, IEEE (2014) 1–6
45. Chen, L., Ali Babar, M., Zhang, H.: Towards an evidence-based understanding of electronic data sources. (2010)

An Analysis Framework for Hybrid Authorship Verification

Seifeddine Mechti [1], Maher Jaoua[2] , Rim Faiz[1,3], and Lamia Hadrich Belguith[2]

[1]LARODEC Laboratory, ISG of Tunis B.P.1088, 2000 Le Bardo, Tunisia
mechtiseif@gmail.com, Rim.faiz@ihec.rnu.tn
[2]ANLP Group, MIRACL Laboratory, University of Sfax, BP 1088, 3018, Sfax Tunsia
[3]IHEC of Carthage, 2016 Carthage Présidence, Tunisia
{maher.jaoua,l.belguith}@fsegs.rnu.tn

Abstract. Given a set of candidate authors for whom some texts of undisputed authorship exist, attribute texts of unknown authorship to one of the candidates is called Author verification. This problem acquired great attention due to its new applications in forensic analysis, e-commerce and plagiarism detection. The author verification task is of great help in the plagiarism detection process. Indeed, the probability of plagiarism increases where two parts of a document are not assigned to the same author. This paper introduces an analysis framework for hybrid authorship verification. In fact, the proposed method takes advantage of a large set of linguistic features to fully address the identification of the document's author. These features are explored to build a machine-learning process. We obtained promising results by relying on PAN@CLEF 2014 English literature corpus.

1 Introduction and Related Works

Although the writing style analysis is an old research area and has been applied successfully to solve many problems, notably authorship attribution, it is obvious that its application to identify the authors of anonymous texts still needs to be investigated further.

Author attribution consists in identifying the author, one of a list, who wrote a particular anonymous text, this categorization focus on open-set[1] or closed-set[2] classification problems [1]. A more difficult author attribution task is the author verification. In this task, we addresses a non-factoid question: "was a particular text written by a well-defined author?".

Recently, the issue of determining the authorship of a document acquired great attention due to its new applications in forensic analysis, plagiarism detection, forensic linguistics, and e-commerce. Additionally, the author identification task is of great

[1] The true author of the disputed text is not necessarily included in the set of candidate authors.
[2] The true author of the disputed text is necessarily included in the set of candidate authors.

help in the plagiarism detection process. Indeed, the probability of plagiarism increases where two parts of a document are not assigned to the same author. Forensic analysis or the analysis of the paternity of documents for legal purposes can contribute to several investigations focusing on various linguistic characteristics.

We grouped methods of authors identification essentially into three categories. The first one is based on a linguistic analysis. The second method is based on various statistical analyses. The more recent third one uses machine learning algorithms.

The basic idea of the stylistic methods is based on the modeling of authors from a linguistic point of view. We cite as an example the works of Li et al. who have focused on topographic signs [2] as well as the works of Zheng et al. who were interested in the co-occurrence of character n-grams [3]. Other works were concerned with the distribution of function words [4] or the lexical features [5]. In another work, Raghavan et al. exploited grammars excluding the probabilistic context to model the grammar used by an author [6]. Feng et al. based their work on the syntactic functions of words and their relationships in order to discern entity coherence [7]. Other studies have focused on the semantic dependency between the words of written texts by means of taxonomies and thesaurus [8].

The first attempts emerged in the studies of [10], constituting the first real great statistical study of texts; they compared the occurrence frequency of words such as verbs, nouns, determinants, prepositions, conjunctions, and pronouns.

In the last few years, a number of new methods which are based on various statistical tools have been presented in order to discriminate between the potential authors of a text. Among these methods, we find inter-textual distance [11], the Delta method [12], the LDA distribution [13] and the KL divergence distance [14].

Recently, from a machine learning point of view [1], author verification method is intrinsic or extrinsic, intrinsic methods use only the known texts and the unknown text of the problem[3] and extrinsic methods uses external documents of other authors for each problem.

The training corpuses are represented in a varied form; we can consider each text as a vector in a space with several variables. In addition, a variety of powerful algorithms can be used to build a classification model, including discriminating analysis [15], SVM [16], decision trees [17], the neural network [4], genetic algorithms [18]. [17] adopt a machine learning approach based on several representations of the texts and on optimized decision trees which have as entry various attributes. This method obtains the first rank in competitive conference Pan@clef2014 only in English essays [1].

The rest of the paper is organized as follows. Section 2 presents the proposed method. Section 3 provides the implementation of the HaiTay System. Section 4 presents experimentations and evaluation. Finally, Section 5 draws our conclusions.

2 Proposed Method

Hybridization has always been considered an interesting track because it overcomes the limitations of combined approaches. It is with this objective in mind that we tried

[3] We call "problem" any test document whose paternity is unknown.

to experiment with learning techniques on all the stylistic and statistical features that have shown their efficiency in the literature. The basic idea is to create for each text T, whose belonging to an author A we want to verify, a sub corpus which includes all the texts written by this author and the texts that are close to it in terms of distance. Thus, if the text was written by author A then there is a high probability that we recognize the style via the stylistic and statistical features of author A's texts belonging to the corpus. On the other hand, if A is not the writer of T then there is a good chance that it is assigned to another author selected from the rest of the sub corpus.

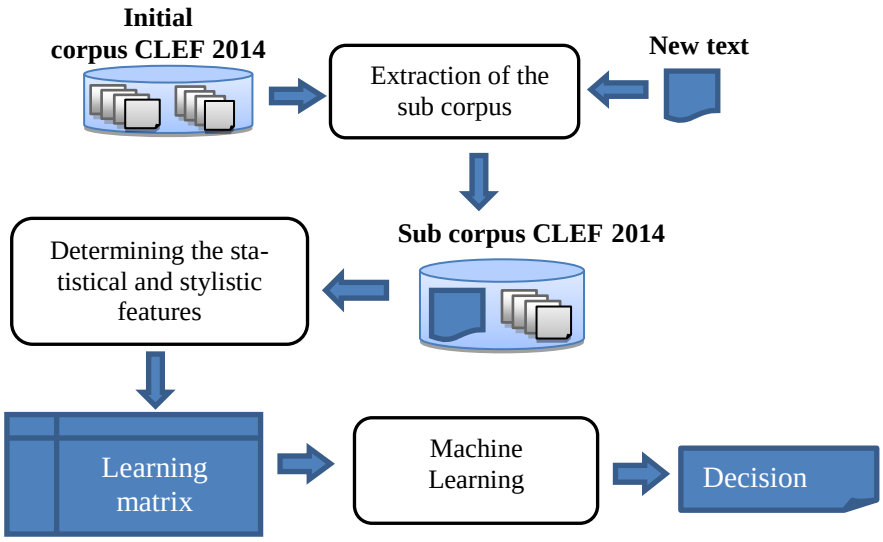


Fig. 1. Steps of the proposed method

3 Implementation of the HyTAI System

In order to implement the proposed method, we developed a system called HyTAI (Hybrid Tool for Author Identification) whose modular decomposition follows the proposed method. Thus, we used the Delta rule in the extraction module of the sub corpus to calculate the distance between two texts. Also, we used the OpenNLP for the extraction of the stylistic and statistical features.

To calculate the distance between two documents, we used the Delta distance proposed by Burrows et al. (Burrows 2002). This distance, which takes into account the most frequent words, is characterized by the following formula:

$$\Delta(Q, A_j) = 1/m \sum_{i=1}^{m} |Zscore(t_{iq}) - Zscore(t_{ij})|$$

where

$$Z\ score\ (t_{tij}) = \frac{tfr_{ij} - mean_i}{sd_i}.$$

Note that tfr_{ij} is the frequency of the term t_i in the document Dj while $mean_i$ is the mean and sd_i is the standard deviation.

It should be noted that if two texts are quite close, then delta tends toward 0. Similarly, the value m may vary from one corpus to another and that is why we conducted an experiment to have the value determined (see next section). For the training sub corpus, we choose the nearest texts of a document to be checked in such a way that a balance is achieved between the texts written by the author to be identified and the texts that do not belong to that author.

In order to extract the stylistic and statistical features, we used tools from the Apache OpenNLP library, which contains a set of functions that can segment texts and perform the syntactic and lexical analyses. We calculated the frequency of lexical features, the ratio V / N – where V is the hapax's size and N is the text length – and the average length of sentences. Regarding parsing, also conducted through the OpenNLP, we extract the number of nouns, the number of verbs, the number of adjectives, the number of adverbs and the number of prepositions.

Then to extract the features related to the model of the language, we consider the text as a simple sequence of characters and determine the frequencies of the letters, the punctuation marks and the numeric characters as well as n-grams.

4 Results Analysis

In this section, we present the experimental results of our method for the identification of authors. We first describe the corpus and the measures of evaluation. Then, we present the performance of our system in the identification of anonymous authors.

Table 1. Dataset description

Number of proposed problems	Number of known documents / author	Average length of unknown documents	Average length of knowns documents
200 problems	2.65 documents	806.86 words	845.30 words

The dataset includes a set of folders from the PAN@CLEF 2014 computational conference. Each folder includes up to five documents and a test document in English. The length of the documents varies from a few hundred to a few thousand words. We should note that we carried out the experiment with the 200 existing problems in the corpus.

In our evaluation, we compare different variants of our proposed stylistic, Statistical n-grams and hybrid author verification methods:

- Stylistic method using lexicals (**le**), syntactic (**sy**), characters (**ch**) and stylistic (**st** = **le**+ **sy** + **ch**) features.
- Statistical method using the Delta rule (**Statis**).

- Machine learning method using SVM, decision table, decision trees, naïve bayes, etc.
- A hybrid method, based on SVM, using both the categories of stylistic features and the Delta rules (**St**+ **Statis** + **Ma**) as described in Ffigure 6;
- A baseline method using n-grams with n = 3, 4, 5, 6 and 7.

The evaluation score c@1[19] has the advantage of taking into account the documents that the classifier is unable to assign to a category. For each problem, each score greater than 0.5 is considered as a positive response and the document is indeed the property of the author in question. Each score below 0.5 is considered as a negative response and therefore the test document does not belong to this author. Nevertheless, all the scores equal to 0.5 correspond to the outstanding problems where the answer will be "I don't know". Then, c@1 is defined as follows:

$$c@1 = (1/n)*(nc + (nu*nc/n)) \quad [19] \tag{3}$$

where n is the number of problems; nc is the number of correct answers; nu is the number of unanswered problems.

The histograms below present the experiments we conducted to obtain the best possible documents paternity.

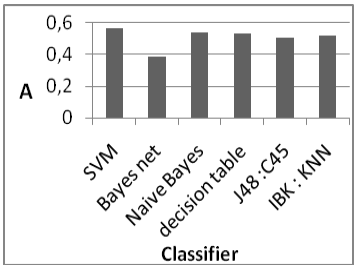


Fig. 2. The accuracy of different classifiers

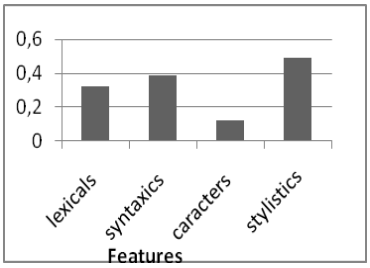


Fig. 3. The c@1 performance of different types of features

Figure 2 shows the accuracy reached with a test set of six well-known classifiers in order to select the best. This is determined with all the stylistic features and the n-gram features (variation of n between 3 and 7). The best accuracy has been achieved by the use of the SVM algorithm with a slight advantage compared to the Naïve Bayes classifier.

Using the SVM classifier, we examine the three categories of features, each category apart and then the 3 gathered categories.

The result presented in Figure 3 shows that the character features are not very powerful in determining the authors of documents whose origin is unknown. On the other hand, the syntactic features give encouraging results. The combination of these features provides a better performance than the use of each feature alone.

Figure 4 depicts the c@1 histogram of the n-grams method. This figure shows that accuracy reaches a maximum for n= 3 and 4, and then it decreases with the increase of n. Therefore, the n-gram models reach a good performance between 3 and 4, and

then they will not be effective. Then, we use the most frequent number of m words between 100 and 400.

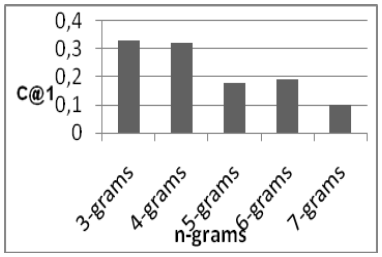


Fig. 4. The c@1 performance according to the n-grams methods

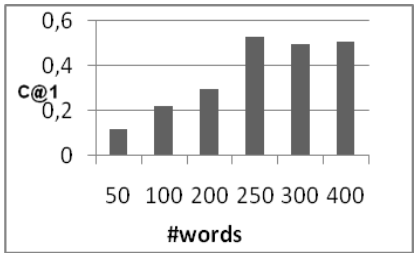


Fig. 5. The c@1 performance according to the number of words

Figure 5 shows that the best c@1 measure is obtained based on the SVM algorithm with 250 words. Then it decreases with the increase of number of words.

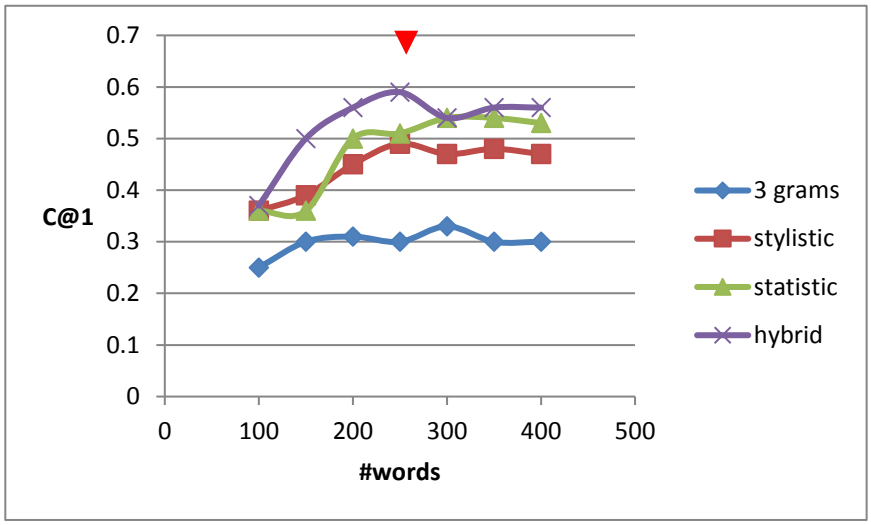


Fig. 6. The C@1 Performance of different features according to words number

Figure 6 shows that the combination of the syntactic features, the lexical ones and the 3-grams brings an encouraging result in a machine learning process. However, the use of delta method for the classification of documents gave better results than stylistic method, we obtain 0.54 c@1 score.

In the hybrid evaluation set up, this result is somewhat improved by using the Delta method. These measures reach a very good value with the choice of the most frequent 250 words. Our system has proven its effectiveness when the statistical and the stylistic analysis are combined. We have been able to find the unknown author of a document in 59% of cases.

Based on Table 2, we compared the performance of our method with those of the winner of PAN@CLEF 2014 competitive conference for the English essays.

Table 2. Performances of our method in comparison with Frery et al.

	Our method	Frery et al. [17]
C@1	0.59	0.71
AUC[4]	0.6	0.72

Our classification, compared with the best systems, is encouraging, which shows the effectiveness of our method. With C@1 equal to 0.59 we obtain the 4th Rank.

5 Conclusion

In this study, we built a hybrid method by combining linguistic features and n-grams. Through experiments relying on a real-world corpus, we showed that the hybrid method outperforms some other methods since we combine syntactic features, lexical features, n-grams and character features. This demonstrates the great potential of heterogeneous models in detection of document's paternity.

The experiments described in this paper were performed on Pan@CLEF 2014 corpora comprising documents in English. We obtained comparable results to the best performing systems

Our method best configuration is 3 as the n-grams length, only 250 as the number of terms and SVM as the learning algorithm.

As future work, we seek to improve our method using a text-extraction tool. We aim to introduce the idea that the style of the author resides in one part of the document rather than in others.

References

1. Stamatatos, E., Daelemans, W., Verhoeven, B., Potthast M., Stein, B., Juola P., Sanchez-Perez, M., Barrón-Cedeño, 2014. A.: Overview of the Author Identification Task at CLEF. England
2. LI, J., Zheng, R., Chen, H. 2006. From fingerprint to writeprint. Communication ACM 49(4), 76-82.
3. Zheng, R., Li, J., Chen, H., and Huang, Z. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology, 57(3), 378-393.
4. Vartapetiance, A., Gillam, L.2014. A Trinity of Trials: Surrey's 2014 Attempts at Author Verification. Proceedings of PAN@CLEF.
5. Argamon, S., Whitelaw, C., Chase, P., Hota, S.R., Garg, N., Levitan, S.:2007. Stylistic text classification using functional lexical features Journal of American society of information science and technology 58(6), 802–822.

[4] Area under the roc curves [1]

6. Raghavan, S., Kovashka, A., Mooney, R.2010. Authorship attribution using probabilistic context-free grammars .Proceedings of ACL'10, 38–42.
7. Feng, V.W., Hirst, G..2013. Authorship verification with entity coherence and other rich linguistic features. Proceedings of CLEF'13.
8. Mccarthy, P.M., Lewis, G.A., Dufty, D.F., Mcnamara D.S.. 2006. Analyzing writing styles with coh-metrix.Proceedings of FLAIRS'06, 764-769.
9. Baayen, R.H. 2008.Analyzing Linguistic Data.: A Practical Introduction to Statistics using R".Cambridge, Cambridge University Press, Cambridge.
10. Mosteller, F., Wallace, D. 1964. inference in an Authorship Problem, In Journal of the American Statistical Association, Volume 58, Issue 302, 275-309.
11. Labbé, C. 2003. Inter-Textual Distance and Authorship Attribution. Corneille and Molière, In: Journal of Quantitative Linguistics, 213-231.
12. Burrows, J.: Delta: 2002. A Measure of Stylistic Difference and a Guide to Likely Authorship, In Journal Lit Linguist Computing.
13. Blei, D.M., Jordan, M.I. 2004 Variational methods for the Dirichlet process. In Proceedings of the twenty-first international conference on Machine learning ACM.
14. Hershey, J.R., Olsen P.A., Rennie, S.J. . 2007. Variational Kullback-Leibler divergence for Hidden Markov models. IEEE Workshop on Automatic Speech Recognition and Understanding.
15. Stamatatos, E., Fakotakis, N., Kokkinakis, G.. 2000. Automatic text categorization in terms of genre and author, Computational Linguistics, Volume 26,.471-495.
16. Lee, C., Mani, I., Verhagen, M., Wellner, B., Pustejovsky, J.: 2006.Machine learning of temporal relations". In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. 753-760.
17. Frery J, Largeron ch, and Juganaru-Mathieu, M. 2014. UJM at CLEF in Author Identification. PAN@CLEF2014.England.
18. Moreau, E., Jayapal, A., Vogel, C. 2014. Author Verification: Exploring a Large set of Parameters using a Genetic Algorithm. Notebook for PAN at CLEF 2014. England.
19. Peñas, A. and Rodrigo, A. 2011. A Simple Measure to Assess Non response. In Proc. Of the 49th Annual Meeting of the Association for Computational Linguistics, Vol.1, 1415-1424.

Reviewing Committee of the Volume

Ajith Abraham
Bayan Abushawar
Hanady Ahmed
Yeşim Aksan
Hend Alkhalifa
Adil Alpkocak
Rania Al-Sabbagh
Galia Angelova
Marianna Apidianaki
Aladdin Ayesh
Alexandra Balahur
Sivaji Bandyopadhyay
Leslie Barrett
Roberto Basili
Anja Belz
Christian Boitet
Igor Bolshakov
Karim Bouzoubaa
Cem Bozsahin
Nicoletta Calzolari
Nick Campbell
Fazli Can
Michael Carl
Violetta Cavalli-Sforza
Niladri Chatterjee
Ilyas Cicekli
Dan Cristea
Samhaa El-Beltagy
Michael Elhadad
Gülşen Eryiğit
Aly Fahmy
Anna Feldman
Alexander Gelbukh
Dafydd Gibbon
Roxana Girju
Gregory Grefenstette
Ahmed Guessoum
Tunga Gungor
Hatem Haddad
Kais Haddar
Lamia Hadrich Belguith
Eva Hajicova
Sanda Harabagiu
Yasunari Harada
Ales Horak
Nancy Ide
Diana Inkpen
Aminul Islam
Guillaume Jacquet
Miloš Jakubíček
Doug Jones
Sylvain Kahane
Bahar Karaoglan
Dimitar Kazakov
Imtiaz Khan
Alma Kharrat
Tarik Kisla
Philipp Koehn
Leila Kosseim
Senem Kumova Metin
Mathieu Lafourcade
Bing Liu
Cerstin Mahlow
Suresh Manandhar
Diana Mccarthy
Alexander Mehler
Farid Meziane
Rada Mihalcea
Evangelos Milios
Ruslan Mitkov
Dunja Mladenic
Hermann Moisl
Masaki Murata
Preslav Nakov
Costanza Navarretta
Nicolas Nicolov
Joakim Nivre
Kjetil Nørvåg
Attila Novák
Nir Ofek
Farhad Oroumchian
Partha Pakray
Ivandre Paraboni
Saint-Dizier Patrick
Maria Teresa Pazienza
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Soujanya Poria
Marta R. Costa-Jussà
Ahmed Rafea
Allan Ramsay
Fuji Ren
German Rigau
Fabio Rinaldi
Horacio Rodriguez
Paolo Rosso
Vasile Rus
Franco Salvetti
Rajeev Sangal
Kepa Sarasola
Fabrizio Sebastiani
Nasredine Semmar
Khaled Shaalan
Altynbek Sharipbayev
Serge Sharoff
Bernadette Sharp
Grigori Sidorov
Kiril Simov
John Sowa
Efstathios Stamatatos
Dzhavdet Suleymanov
Maosong Sun
Jun Suzuki
Stan Szpakowicz
Hristo Tanev
William Teahan
J.-M. Torres-Moreno
George Tsatsaronis
Olga Uryupina
Manuel Vilares Ferro
Aline Villavicencio
Piotr W. Fuglewicz
Marilyn Walker
Andy Way
Bonnie Webber

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
abril de 2016
Printing 500 / Edición 500 ejemplares