Exploring the Partial Textual Entailment Problem for Bengali News Texts

Amitava Das[1] and Dwijen Rudra Pal[2]

[1] Department of Computer Science and Engineering,
University of North Texas, Denton, Texas,
USA

[2]Computer Science and Engineering Department,
National Institute of Technology, Agartala,
India

Abstract. Formal definition textual entailment implies strict meaning relationship of meaning in its totality between text (*T*) and hypothesis (*H*). Even if the text have main ideas of a hypothesis, but lacks minor information or have additional information then treated as an entirely unrelated text. In these cases we are left with no sense of how close (*T, H*) were to entailment. In various applications of entailment, major attention has given on strict entailment only. However, in reality strict entailment cases are relatively lower in compared to partial entailment cases, are prevalent. We introduce the idea of partial entailment in this paper and defining it empirically. We have developed corpus and finally proposed baseline architecture for automatic identification of partial textual entailment. This work is on Bengali news texts.

Keywords: Partial textual entailment, Bengali news texts.

1 Introduction

The automatic recognition of textual entailment is one of the difficult under research Natural Language Processing (NLP) problems. In the last decade, automatic textual entailment research received significant research attentions but majority of the works dealt with strict definition of entailment, whereas in reality strict entailment cases are relatively lower. Moreover, majority of such prior works concentrated on English. Here in this paper we are proposing the idea of partial textual, as a bidirectional relationship between pairs of statements for Bengali. The standard definition of textual entailment has been considered as a unidirectional problem so far where a given text *T* would be considered as entailed to another text *H* (hypothesis) if the meaning of the *T* could be completely inferred from the *H*. We are extending the formal definition of the entailment and empirically defining the concept of partial entailment. Let us consider "*Text1*" or *(T1)* and "*Text2*" or *(T2)* are partially entailed with each other. Both the statements *T1* and *T2* would be considered as entailed to

each other while partial meaning of *T1* could be inferred from the partial or complete meaning of *T2* or vice versa. We are also proposing different classes of partial entailment by breaking down both the statements *T1 and T2* into additional information to compare partial matching. We preserve the original definition of the complete entailment. Our main motivation for this work was to investigate the idea of partial textual entailment, and assess credibility of existing automatic complete textual entailment detection methods for the partial entailment problem, otherwise finally explore for new methodologies.

The rest of the paper is organized as follows. In the next sections, we formalize our procedure by empirical definition of partial entailment in Section 2, Corpus Acquisitioning in Section 3, a baseline system and performance in Section 4, preparing related work on partial textual entailment in Section 5, and finally, we draw our conclusions in Section 6.

2 Partial Entailment: The Empirical Definition

We define following four detailed categories to represent partial entailment.

1. **Type1:** If both the sentences are having same information and meaning same, then it is a case of direct entailment and should be noted as YES (X=Y). This category is the perseverance of the original entailment definition. Example:

   Sentence 01: মৃত্যুদণ্ড নয় যাবজ্জীবন আফতাবের।

   Eng. Gloss: Aftaab has been life sentenced instead of sentence to death.

   Sentence 02: ফাঁসি রদ করে আফতাবকে যাবজ্জীবন দিল সুপ্রিমকোর্ট।

   Eng. Gloss: Supreme Court has cancelled aftaab hanging and had given him life sentenced

   Entailment Status: *YES (X=Y).*

2. **Type2:** If the second sentence has all the information of the first sentence and has some extra information, then it is a case of partial entailment of type1 (X=X+Z). System also marked the repeated information section in the second sentence. Example:

   Sentence 01: (ফাঁসি থেকে রেহাই পেয়েছে এই ঘটনার আর এক দোষী জামিলউদ্দিননাসির।)

   Eng. Gloss: Jamiluddin Nasir, another offender of this incident is exempted from hanging

   Sentence 02: আজ দেশের সর্বোচ্চ আদালত আফতাব ও তার সঙ্গী (জামিলউদ্দিন নাসিরের ফাঁসি রদ করল।)

   Eng. Gloss: Today Supreme court has cancelled hanging of Aftaab and his companion Jamiluddin Nasir

   Entailment Status: YES (X=X+Z)

3. **Type3:** If the first sentence has all the information of the second sentence and has some extra information, then it is a case of partial entailment of type3 (X+Z=X+Y). Moreover please mark the repeated information section of the first section. Example:

   Sentence 01: দীর্ঘ শুনানির পর (নিম্ন আদালত দোষীদের মৃত্যুদণ্ড ঘোষণা করে।)

   Eng. Gloss: After a long hearing lower court has declared sentenced to death of the offenders.

   Sentence 02: জঙ্গিদল আসিফ রেজা কমান্ডো ফোর্সের সদস্য আফতাব জামিলউদ্দিন সহ সাতজনকে (নিম্ন আদালত মৃত্যুদণ্ড দেয়।)

   Eng. Gloss: Lower court has declared sentence to death to Seven members of the terrorist group of Asif reja force along with Aftaab, Jamiluddin.

   Entailment Status: YES (X+Z=X+Y)

4. **Type4:** If both the sentences are not having same information then it is a false entailment and marked NO status. Example:

   Sentence 01: ফাঁসি থেকে রেহাই পেয়েছে এই ঘটনার আর এক দোষী জামিলউদ্দিন নাসির।

   Eng. Gloss: Jamiluddin Nasir, another offender of this incident is exempted from hangingSentence 02: প্রায় এক বছর চলা শুনানি শেষ হয় এমাসেই।

   Eng. Gloss: Almost one year hearing has finished in this month

   Entailment Status: NO

Here in all the cases X, Y and Z are abstract representation of a block of information. The 4th category is basically the negative example. We marked the common information boundaries for all the sentence pairs. For our automatic partial entailment detection task we prefer to detect common information boundaries for the both sentences beyond the original binary classification, because then it will be useful to use those outputs further for any NLP task like summarization, QA, or else. The empirical question we asked to ourselves is how much extra information should be the upper limit for the partial entailed sentence pairs. For example we cannot claim that the following two sentences are partially entailed.

Sentence 01: গত তিন দিন ধরে চলা যাবতীয় জল্পনার অবসান ঘটিয়ে পাকিস্তান আজ জানিয়ে দিল সোমবার নরেন্দ্র (মোদীর শপথগ্রহণ অনুষ্ঠানে যোগ দিতে আসছেন নওয়াজ শরিফ)।

Eng. Gloss: Over three days, ending speculation Pakistan confirmed that Nawaz Sharif would attend the oath ceremony of Narendra Modi on Monday.

Sentence 02: মোদির শপথে নওয়াজ শরিফ।

Eng. Gloss: Nawaz Sharif at Modi's oath ceremony.

Entailment Status: NO

Here in the first sentence there is lots of more information than the second one. So, we define out upper threshold as the following equation.

$$\frac{\left|n_1^w - n_2^w\right|}{(n_1^w + n_2^w)/2} * 100 \pounds 35\%$$

Here n_1^w is the total number of words in the sentence one and n_2^w is the total number of words in the second sentence. In our definition of partial entailment we kept the number of words differences within 35%. To compare we checked the mean word count difference in the standard RTE (Recognizing Textual Entailment) corpus[1] and we found it is to be 75-80% on an average. So, empirically we have restricted more than 2 times for of the original textual entailment definition.

3 Corpus Acquisition

We designed a semi-automatic corpus acquisition process, because it helps on removing rigorous manual efforts and expedite the overall process. We collected news texts on specific events from multiple Bengali news sources, i.e. news stories on same event published in different newspapers on the same day. Two most popular Bengali newspapers Aajkaal[2], and Anandabazar[3] are chosen for this task. During the selection of the source texts, we gathered Bengali news text corpus of 25 topics containing news stories on those events in the cited two newspapers. From the original HTML text we kept only the unformatted content text, without any images, tables or links. Further, while choosing topics we made sure those topics covering various domains like international politics, national politics, sports, natural disasters, political campaigns and elections.

Here from the next paragraph onwards various steps of automatic semi-automatic annotation task have been discussed. We have also included some useful negative examples, are lexically very similar but not actually entailed.

3.1 Stop Word Removal

Stop/junk words such as *অবশ্য (sure)*, *অনেক (many)*, *অন্তত (at least)*, *অথবা (or)*, *অথচ (still)*, *আজ (today)* are removed automatically. The stop word list for Bengali has been collected from [1].

3.2 Tokenizing and Part-of-Speech (POS) Tagging

A tokenizer has been developed for Bengali text. The Bangla POS-Tagger, developed by [2, 3] has been used for the present task.

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE/Datasets
[2]http://www.aajkaal.net/
[3]http://www.anandabazar.com/

3.3 Stemming

Stemming is the process of generating surface word forms to their root forms. For example, the plural forms of a noun such as 'সেন্টারের' (center's) are stemmed into 'সেন্টার', (Center) and 'আফতাবের' (Aftab's) are stemmed into 'আফতাব' (Aftab) for the present task. Some of the most frequent Bengali suffixes are 'ই', 'গুলো', 'টা', 'টি', 'রা', 'হীন'. We have used the system described in the [4], with some simple additional modifications.

3.4 Content Words Extraction

At this stage bag of content words have been collected from each sentence to further measure cosine similarity between sentences. Here bag of content words defined [5] by only four open POS classes: nouns, verbs, adverbs and adjectives. The used POS tagger [4] generates two sub-categories for Noun; Verb has two sub-categories as verb finite and verb auxiliary. Adverb and Adjective does not have any more subcategories.

3.5 Measuring Cosine Similarity

The simplest way to describe a binary sentence vector is as the set of its non-zero values. Cosine similarity is a measure of similarity between two *n*-dimensional vectors obtained by finding the cosine of the angle between them. It is often used to compare documents in text mining. In addition, it is used to measure cohesion within clustering data mining. Cosine similarity is also widely used in information retrieval to calculate the similarity between documents or sentences. Given two vectors of attributes, *A* and *B*, the cosine similarity θ is calculated using the dot product and magnitude as:

$$\cos(A, B) = \frac{|A \cup B|}{\sqrt{|A| \times |B|}} \quad (1)$$

We consider binary vectors, that is, vectors with entries that are either 0 or 1. We converted each sentences into binary vector. Then calculates the cosine similarity for all the sentences present in file 01 with other files within the same topic cluster. One example given below to show the similarity value. These lines are from the original texts after stemming of content words. For example,

sentence 1: মৃত্যুদণ্ড আফতাব যাবজ্জীবন,

sentence 2: আফতাব যাবজ্জীবন দেওয়া হয়েছ,

Cosine Similarity Score (1,2) = 67.082.

Then we ended up with various sub-groups of possible partial entailed pairs. For further manual checking, we chose a cosine similarity threshold of $^{3}15$ experimentally. It has found that almost all the actual entailment cases where cosine similarity value is less than the 15 of maximum cosine similarity value, no entailment relation comes.

3.6 Manual Annotation and Agreement

For the human annotation we involved two different human annotators, they are undergraduate students (not linguist) and native Bengali speakers. To assess annotation agreement Cohen's Kappa [6] coefficient has been measured on a small subset. We have chosen one topic: two files having 144 comparisons altogether tagged by the two annotators separately. A detailed categorical distribution of the two annotator's markings is reported in the following Table 1.

Table 1. Categorical distribution for the agreement annotation.

	Categories			
	X=Y	**X=X+Z**	**X+Z=X+Y**	**NO**
Annotator 01:	4	7	4	129
Annotator 02:	5	8	2	129

We found the sentence level *kappa* is 0.92. To understand the common information boundary detection agreement we choose Mean Agreement precision (MAP) metric. For the Type 2 (X=X+Z) it is 0.98 and for the Type 3 (X+Z=X+Y) it is 0.976, which is indeed higher implies that the task is not much ambiguous.

3.7 Corpus Statistics

Finally we ended up with 245 tagged pairs, it is an ongoing task. Here are the details of corpus statistics. All the negative examples, marked as not entailed by the annotators are been kept for further evaluation during training and testing. For the negative example inclusion cosine similarity threshold is $^{3}10$. A detail of the generated corpus is reported in the Table 2. This is an ongoing task.

Table 2. Categorical distribution for the agreement annotation.

Categories	**Sentence Pairs**	**Avg. CS**
X = Y	102	54.68
X=X+Z	127	16.41
X+Z=X+Y	16	23.12
Neg. Exmp.	7,349	10.19

4 The Baseline System and Performance

At this stage our motive is to develop an automatic system, can identify partial textual entailment (PTE) and can classify them into defined classes. We have developed a very basic system and the accuracy is not very encouraging but we are reporting the initial results to establish the fact that the partial entailment detection is more challenging than the standard definition of the entailment. This is an ongoing task.

Pakray et al. (2011) [10] reported decent performance for their rule based automatic textual entailment system using lexical and syntactic features. Reported lexical features were WordNet based Unigram Match, Bigram, Longest Common Subsequence (LCS), Skip-grams and they stemmed throughout before each of the feature compilation. Syntactic features were Subject, Object, Noun, Verb, Preposition, Determiner and Number. We drew our inspiration from this task and applied those lexical features on our data to observe the effect. We are unable to use syntactic features because there is no good quality dependency parser available for Bengali. Syntactic features extraction is our future target.

There is no WordNet (Bengali) available publicly so we are unable to use that feature. Therefore, we did our experiment with only Unigram Match, Bigram, Longest Common Subsequence (LCS), Skip-grams and we have used stemming before each feature extraction. All the features are self explanatory except Skip-grams. A skip-gram is any combination of n words in the order as they appear in a sentence, allowing arbitrary gaps. In the present work, only 1-skip-bigrams are considered where 1-skip-bigrams are bigrams with one word gap between two words in order in a sentence. Our strategy is relatively simple. Pakray et al, 2011 [10] reported their accuracies on the RTE 1-5 datasets as in the following Table 3.

Table 3. Pakray et al. (2011) reported RTE accuracies.

Dataset	Accuracy	Baseline PTE
RTE1	0.537	0.49
RTE2	0.592	0.51
RTE3	0.610	--
RTE4	0.554	--
RTE5	0.603	--

Pakray and his colleagues did not mention any implementation details how these features helped on the final entailment decision and how all these features values accumulated to reach out the final result. Moreover they did not provide any feature ablation to understand what is the effect of a particular feature. We replicated the system using these formulations.

Table 4. Lexical features meanings.

Type	Unigram	Bigram	LCS	SkipGram
	$(u_m/n)*100$	$(b_m/n)*100$	$(lcs_m*l^{avg}/n)*100$	$(sg_m*l^{avg}/n)*100$
X = Y	52	33	16	20
X=X+Z	14	9	4	10
X+Z=X+Y	21	12	9	14
Neg. Exmp.	10	8	2	6

where u_m is the total number of matched unigrams and n is the average number of words in both the sentences. b_m is total number of matched bigrams. lcs_m is the numbers of LCS matched whereas l^{avg} is the average length of those matched strings. sg_m is the numbers of skip-gram matched and l^{avg} is the average length of those skip-grams. Reported numbers are the mean values of those features learned from the training set.

Table 5. Baseline PTE with basic lexical features.

Type	Accuracies
X = Y	0.47
X=X+Z	0.39
X+Z=X+Y	0.35
Neg. Exmp.	0.56

We split our data into training (65%) and test set (35%). This split was class wise. Those learned feature wise mean values have been used further to detect partial entailment classes on the test set. Feature values exceeding these means resulting *yes* decision and feature values below the means is resulting a *no* decision. Initial results reported in the following table 5. We even tried the same setup on RTE 1 and 2 data as reported in the last column in the table 3.

5 Related Works

The concept of the partial textual entailment was first presented by Nielsen and his colleagues [7] in the year of 2009. Their work was on student's responses to an automated tutor's question. Partial entailment was used to understand the overlap between student answers. To detect proposed system broke sentences into fine-grained semantic facets, derived roughly from syntactic dependencies, and checked whether those facets were overlapping. Their work provides a finer-grained annotation schema to indicate more precisely the entailment relationship between the student's answers and that facet of the reference answers.

Instead of binary textual entailment decision in the form of yes or no, the proposed method in Nielsen et al work break down reference answer into semantic facets which refer to some part of a text's meaning. They also propose more expressive annotations labels in order to specify entailment relationship more clearly. They used eight finer annotation labels named: Assumed (facets that are assumed), Expressed (Facet that are directly expressed), Inferred (Facets inferred), Contra-Expr (Facets directly contradicted by negation), Contra-Infr (Facets contradicted by pragmatics), Self-Contra (Facets that are contradicted and implied), Diff-Arg (Facets where core relation expressed) and Unaddressed (Facets not addressed at all). In this model of facets, where each such facet is a pair of words in the hypothesis and the direct semantic relation connecting those two words. In comparison, we identified

common information between *T* and *H* in terms of semantic similarity, which defines semantic inference more precisely for the sake of partial entailment.

Agirre et al. explicitly defined in their work [8] in 2012, different levels of semantic text similarity between two sentences. This system proposed 5 levels of similarity starting from 0 to 5. Level 0 defines no similarity, 1 defines not equivalent but same topic, 2 defines not equivalent but share same details, 3 defines roughly equivalent with missing of important information, 4 as mostly equivalent but some unimportant information differ and 5 as completely equivalent having same information. Though this model provides finer grained similarity notions, it is still not appropriate for semantic inference, as similarity was not well defined enough.

After these works, there is no more work on partial textual entailment until Omer Levy et al work [9] published last year i.e. 2013. In this work, they investigate the idea of partial textual entailment, and assess whether existing complete textual entailment methods can be used to recognize it. In their work partial textual entailment has defines as breaking down the hypothesis into components, and attempting to recognize whether each one is individually entailed by text. This definition concentrated on whether a single element of the hypothesis is entailed or not.

In our work, we proposed two detailed categories of partial entailment with further identification of common information boundaries in both the entailed sentences, which is a first approach in the area of partial textual entailment. This identification will be helpful for any NLP task like summarization, QA.

6 Conclusion and Future Work

In conclusion, we would like to mention that defining various classes of partial textual entailment is the main contribution of this task. Research works on partial entailment is an untouched paradigm so far. Moreover, with best of our knowledge this is the first paper discussing about the entailment problem for the Bengali.

This is an ongoing task. We are collecting more data and experimenting various automatic processes for the partial entailment detection. We are also applying same setup on social media text i.e. tweets.

References

1. http://www.isical.ac.in/~clia/resources.html.
2. A. Senapati, U. Garain: GuiTAR-based Pronominal Anaphora Resolution in Bengali, ACL, pp. 126–130 (2013)
3. http://www.isical.ac.in/~utpal/resources.php
4. Dolamic, L., Savoy, J.: Comparative Study of Indexing and Search Strategies for the Hindi, Marathi and Bengali Languages. ACM Transactions on Asian Language Information Processing, 9(3) (2010)
5. Winkler, E.G.: Understanding Language. Continuum. pp. 84–85 (2007)
6. Cohen J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1):37–46 (1960)

7. Rodney D Nielsen, Wayne Ward, James H Martin. Recognizing entailment in intelligent tutoring systems. Natural Language Engineering, 15(4):479–501 (2009)
8. Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre: SemEval-2012 Task 6: A pilot on semantic textual similarity. In: Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics, pages 385–393, Montreal, Canada (2012)
9. Omer Levy, TorstenZesch, Ido Dagan, Iryna Gurevych: Recognizing Partial Textual Entailment. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL, Volume 2: Short Papers), pages 451–455, Sofia, Bulgaria (2013)
10. Partha Pakray, Alexander Gelbukh, Sivaji Bandyopadhyay: Textual Entailment using Lexical and Syntactic Similarity. International Journal of Artificial Intelligence and Applications, Vol. 2, No. 1, DOI 10.5121/ijaia.2011.2104 , pp. 43–58 (2011)