TRIPLET EXTRACTION FROM SENTENCES USING SVM

*Lorand Dali**, *Blaž Fortuna°*
* Technical University of Cluj-Napoca, Faculty of Automation and Computer Science
G. Bariţiu 26-28, 400027 Cluj-Napoca, Romania
° Jožef Stefan Institute, Department of Knowledge Technologies
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 477 3127; fax: +386 1 477 3315
E-mail: loranddali@yahoo.com, blaz.fortuna@ijs.si

ABSTRACT

In this paper we present a machine learning approach to extract subject-predicate-object triplets from English sentences. SVM is used to train a model on human annotated triplets, and the features are computed from three parsers.

1. INTRODUCTION

As described in [1][2][3] a triplet is a representation of a subject-verb-object relation in a sentence, where the verb is the relation. In [3] triplet extraction methods based on heuristic rules have been described. In this paper a machine learning approach using SVM is tried. The data comes from triplet annotations made by linguists on the Reuters news article corpus. First the triplet extraction method using SVM is presented, then the evaluation method and the results, and finally the conclusions are drawn.

2. EXTRACTION METHOD

In this section the triplet extraction method using SVM will be explained. First we assume that a model is available and we explain how triplets are extracted from a sentence using that model, and then the method for training the model will be explained.

The triplet extraction process is depicted in Figure 1. The input is a sentence, *'The increase will cover all kinds of wheat including durum and milling wheat.'*, in our example. The sentence is tokenized and then the stop words and punctuation (which are grayed out in the picture) are removed. This gives us a list of the important tokens in the sentence, [*increase*, *cover*, *kinds*, *wheat*, *including*, *durum*, *wheat*]. The next step is to get all possible ordered combinations of three tokens from the list. In our case, as there are 8 tokens, we obtain $336 = 8 \cdot 7 \cdot 6$ such combinations, but due to lack of space only 8 of them are shown in the picture. In what follows we shall call these combinations *triplet candidates*. From now on the problem is seen as a binary classification problem where the triplet candidates must be classified as positive or as negative. The SVM model assigns a positive score to those candidates which should be extracted as triplets, and a negative score to the others. The higher the positive score, the 'surer' it is that the triplet candidate is in fact a correct triplet. On the right side of the image in Figure 1eight triplet candidates ordered descending based on their classification scores are shown. The negative ones are grayed out. From the positive ones the resulting triplet is formed. It can be seen that for all positively classified candidates the subject is *increase* and the verb is *cover*, so the first two elements of the triplet are settled. As opposed to the subject and the verb, the objects are different among the positively classified triplet candidates. In such cases an attempt to merge the different triplet elements (in this case objects) is made. The merging is done in such a way that if two or more words are consecutive in the list of important tokens, then they are merged. In our example it was possible to merge all different objects into a single one, and the triplet (*The increase, will cover, all kinds of wheat including durum and milling wheat*) was obtained. The tokens which were obtained from the positive triplet candidates are underlined. Where merges have been done (in the object) the tokens are connected by the stop words from the original sentence. In all cases before the leftmost token all the stop words which come before it in the original sentence are included. Of course, in the merging method described above, it will not always be possible to merge all tokens into a single set. In this case several triplets (one for each of the three sets) will be obtained. An important note which has to be made is that in practice in the classification described above there are many false positives, so it does not work to take them all for the resulting triplets. Instead only the top few from the descending ordered list of triplet candidates are taken (more on how many is in the section describing the results)

3. TRAINING OF THE SVM MODEL

In the previous section describing the triplet extraction method it was assumed that an SVM model is available. Here the training of that model and the features taken into account in the classification of the triplet candidates are presented.

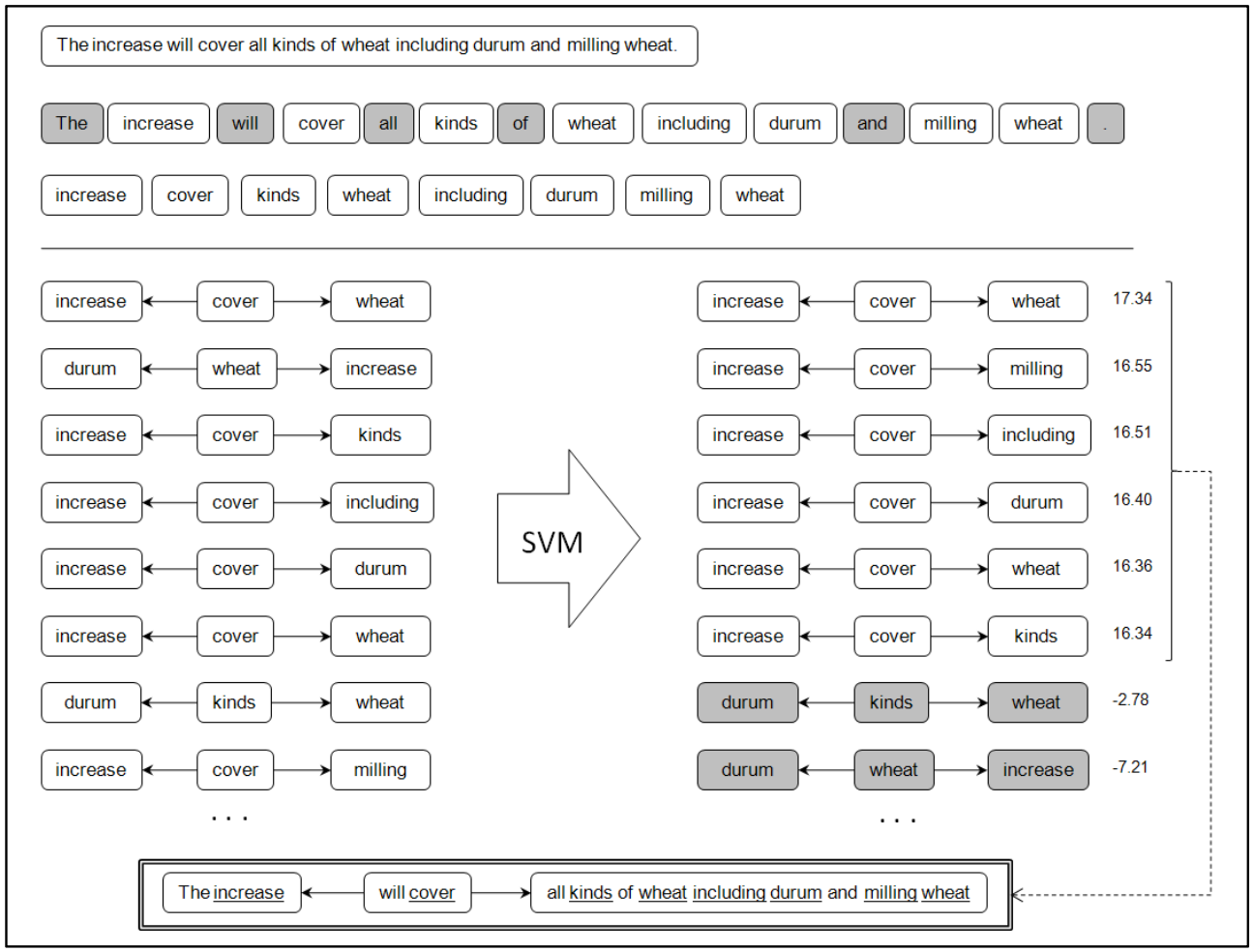

Figure 1 Triplet Extraction Process

The training data comes from human annotated triplets from the Reuters news article corpus. To train the model, from each sentence the triplet candidates are obtained and over 300 features are computed for them. The features can be grouped into the following categories:

- Features depending on the sentence (e.g. length of the sentence, number of stop words etc)
- Features depending on the triplet candidate (e.g. subject, verb and object candidate words, order, subject-verb token distance, context of verb, etc.)
- Features depending on the Treebank parse tree of the sentence (e.g. depth of tree, depth of subject, part of speech of the candidate elements)
- Features depending on the Linkage of the sentence obtained by LinkParser (e.g. number of link types, number of left links from the object etc.)
- Features obtained by the Minipar dependency parse tree of the sentence (e.g. diameter of the subject subtree, category and relation of the uncle of the verb etc.)

The top twenty features in a ranking obtained by information gain are shown below:

1	verb candidate word	11	subj left context word1
2	verb left context word0	12	is the candidate ordered?
3	verb right context word0	13	subj left context word0
4	verb left context word1	14	obj left context word1
5	verb right context word1	15	obj left context word0
6	subject candidate word	16	obj right context word1
7	subj right context word0	17	subj-verb distance
8	subj right context word1	18	obj right context word0
9	object candidate word	19	average word length
10	last 2 characters of verb	20	subj-obj distance

A triplet candidate is labeled positive if its subject token is a substring of a human annotated triple in its sentence, and if both its verb and its object are substrings of the verb and of the object of that triplet.

4. EVALUATION METHOD

For evaluation a way of comparing triplets is needed. To just check whether two triplets are identical would penalize too much those triplets which are almost correct. This is why we define a triplet similarity measure.

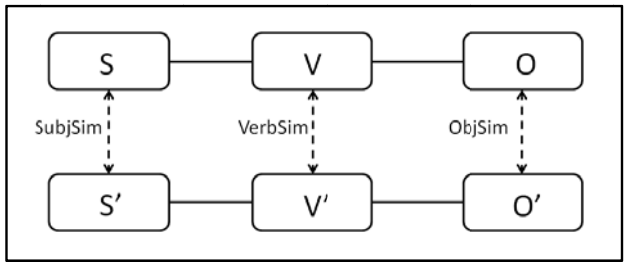

Figure 2 Triplet Similarity Measure

To compute the similarity between two triplets, the subject, verb, and the object similarities are computed and the numbers are averaged. All similarities are real numbers between 0 and 1, where 1 means identity and 0 means no overlap at all. Because the similarity is between 0 and 1 it can be seen as a percentage. As all three elements of a triplet are strings, a string similarity measure is used to compute each of the subject, verb and object similarity measures. The similarity between two strings is obtained by tokenizing an removing the stop words from each of them thus obtaining a collection of tokens for each of the strings. Then we count how many tokens appear in both collections and divide this number by the number of tokens in the larger collection.

Having defined the triplet similarity, we can now compare the triplets extracted with the triplets annotated. For each sentence we have the set of extracted triplets and the set of annotated triplets. We compute the similarity between the corresponding triplets and average the numbers over all sentences. The corresponding triplet of a triplet is the triplet in the other set which is most similar to it. In one of the two triplet sets (depending on the direction of comparison) we can have triplets which have one, more or no corresponding triplets. In the other set each triplet will have exactly one corresponding triplet. We can either compare the extracted triplets to the annotated ones or the annotated triplets to the extracted ones. If we do the first thing we see what proportion of the extracted triplets were annotated (are correct), and we shall consider this proportion the **precision** of the system. If we compare the other way round then we see what proportion of the annotated (correct) triplets have been extracted. We shall consider this proportion showing the **recall** of the system.

5. RESULTS

Applying the methods described previously using a training set of human annotated triplets of 700 sentences and a test set of 100 sentences, a precision of 38.6% and a precision of 46.80% have been obtained. The tables in Figure 5 show how the precision and the recall vary when the training set size and the top proportion of triplet candidates selected as positives are changed. It is apparent that an increased training set size has a positive effect on both precision and recall. A higher proportion of triplet candidates selected as positives increases the recall but deteriorates the precision. We can conclude that it is a good compromise to select the top 1% of the triplet candidates as positives.

Other arguments in favor of choosing the top few from the ordered list of triplet candidates are shown in the histogram in Figure 3 showing how many true positive triplet candidates are on each position in the ordered list. It can be seen that all are in the top 10% in, and the vast majority are in the top 2%.

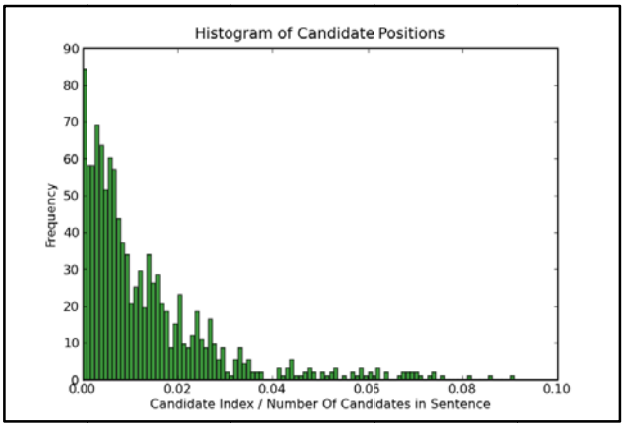

Figure 3 Histogram of the positions of the true positives in the list of ordered triplet candidates

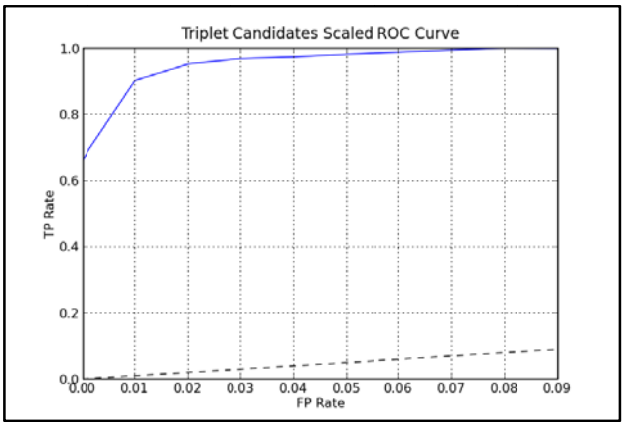

Figure 4 Scaled ROC curve of the triplet candidates

Figure 4 shows the ROC curve with the horizontal axis scaled up 10 times. We can see from it that by selecting

Top Proportion	#1	0.2%	0.5%	1%	2%	3%	4%	5%
Precision(%)	38.14	38.43	**39.70**	38.36	37.23	36.83	36.25	35.67
Recall(%)	29.58	40.23	43.22	46.80	47.29	47.30	47.38	**47.38**

#Train Sent	100	200	300	400	500	600	700
Precision(%)	30.06	36.15	36.55	38.69	35.97	37.37	38.36
Recall(%)	41.64	43.21	44.73	45.14	45.59	46.05	46.80

Figure 6 Influence of training set size and top proportion selection on precision and recall

in such a way that 5% of the negative instances are included in the selection we catch close to 100% of the positives.

It is interesting to see how the different feature types influence the performance. In Figure 6 the first pair of bars is the result of random selection of positive triplet candidates. The second shows the results obtained by evaluating the triplets extracted bay heuristic rules using the OpenNLP parser [3]. In the third bar pair the results of the machine learning approach is shown, but only taking into account features which give part of speech and context information. In the next bars the performance figures obtained by adding the features from diffent parsers incrementally are shown. It can be seen thet by taking into

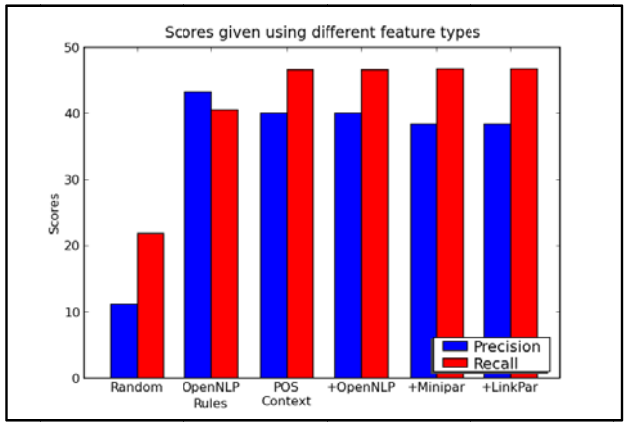

Figure 5 Performance figures obtained by using the different kinds of features

account parsing information the performance does not change significantly.

6. CONCLUSIONS

The conclusions which can be drawn are the following. Although the small size of training and test sets does not allow us to be very conclusive, we can say that the approach presented is promising. The fact that parsing information failed to make a difference in the performance means either that in the small training set of 100 sentences there have been no examples which relied on parsing information to be classified, or that parsing information is not important for triplet extraction. Another issue which slows down the execution is that all ordered combinations of three tokens are considered as triplet candidates. This number increases exponentially with the length of the sentence. For the future the following improvements could be made:

- Building a probabilistic model which will say for a triplet candidate what is the probability of it being a triplet. This would help because as it is now we always select a top ranked proportion, say 5%, on the other hand if we would take those candidates as triplets which have probability more than 95% maybe more and maybe less than a fixed top rank would be selected. We would not assume any connection between the length of the sentence and the number of triplets.
- Solving the classification problem in two phases. First for each word in the sentence we would compute by a probabilistic model how likely it is that the word is a subject, a verb and an object. Having these 3 probabilities for every word we would build triplet candidates where the subject, the verb and the object are correct with a high probability, thus avoiding an exhaustive search of all combinations. This would much decrease execution time.
- Computing only the most relevant features in the classification process

7. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under NeOn (IST-4-027595-IP), SMART (IST-033917) and PASCAL2 (IST-NoE-216886).

References

[1] J. Leskovec, M. Grobelnik, N. Milic-Frayling. *Learning Sub-structures of Document Semantic Graphs for Document Summarization*. In *Proceedings of the 7th International Multi-Conference Information Society IS 2004, Volume B*. pp. 18-25, 2004.

[2] J. Leskovec, N. Milic-Frayling, M. Grobelnik. Impact of Linguistic Analysis on the Semantic Graph Coverage and Learning of Document Extracts, *National Conference on Artificial Intelligence*, 2005.

[3] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenic, *Triplet extraction from sentences*, SiKDD 2007