Study of feature selection to learn SVM models useful for extracting ACE mentions of relations

Norman Salazar Ramirez
Benjamin Auffarth
Sergi Fernandez Langa
course report for
Natural Language Processing for Massive Textual Data Management
at Universitat Politècnica de Catalunya

January 24, 2007

Abstract

We studied the importance of different features for the extraction of relationships from ACE documents in support vector machine (SVM) learning. Features that we extracted for training and classification were the actual words, bigrams of words, annotations, such as part of speech (POS), semantic roles (SR), dependency parses (DP), and name entities (NEs), in positions around, between, and at the positions of the two argument name entities. Performance with linear, polynomial, RBF kernels were at dismal 20% accuracy levels. A custom kernel, a combination of argument kernel and bigram kernel, yielded 71.7% accuracy and 70% f-measure.

1 Introduction

ACE[1] is a research program, sponsored by the US government, in automatic content extraction technology for Arabic, Chinese, English, and Spanish. One of the ACE 06 subtasks, related to relation extraction consists in the recognition and classification of relation mentions (relations detection and recognition, RDR) within the ACE documents, which will be addressed in this report.

[1] short for **A**utomated **C**ontent **E**xtraction

The entity types are ART, GEN-AFF, ORG-AFF, PART-WHOLE, PER-SOC, PHYS. A relation mention in ACE is a sentence or phrase that expresses the relation between two name entities (NE). Both entities involved in the relation must be mentioned in the sentence (i.e. there is no across-sentence processing). The sentence "Peter and his daughter, Jane, will go to the cinema" is a relation mention of the type "family" involving two entity mentions "Peter" and "Jane". Some relations are symmetric (e.g. "family"), some assymetric (e.g. $< GeorgeBush, France >$ as "physical located relation"). Relation output is required for each document that mentions the relation and has to include the following: (NIST, 2006)

- attributes of the relation: type, subtype, modality, tense.
- relation arguments: ID and role. Optionally one or more temporal arguments (timex2s)
- relation mentions: Sentence or phrase within which the relation is mentioned.

Evaluation for the RDR task is based on entities, temporal arguments, and values. The English resources of the ACE system training and evaluation corpora, respectively, consist of the following sources: (NIST, 2006)

Source	**training size** (10^3)	**evaluation size** (10^3)
Broadcast News	60	10
Broadcast Conversations	45	7.5
Newswire	60	10
Weblog	45	7.5
Usenet	45	7.5
Conversational Telephone Speech	45	7.5

The objective of this report consists in comparing different feature sets and different kernels for the classification of pairs of NE mentions occurring in the same sentence into an ACE relation type:subtype. Classification of relationships between name entities is mostly restricted to types (as opposed to type:subtype) and we did not try to classify subtypes after initial rather futile attempts.

Relation extraction for ACE06 was performed previously by (Zhao & Grishman, 2005) and (Zhou, Su, Zhang, & Zhang, 2005). Their research indicated the importance of a well-engineered feature space. Both papers concluded that lexical information, sentence ordering, and chunking kernels have the highest contribution to the overall performance of a model for relation extraction.

(Zhou et al., 2005) used lexical features, entity type features, mention level features (name, nominal, pronoun), chunking features, dependency features, and

full parsing features. Their analysis found entity types, chunking, semantic resources (wordnet) to have the highest contribution to relation extraction. Almost useless were dependency and full parsing information.

(Zhao & Grishman, 2005) combined features using kernels in binary-class SVM and k-NN. They used different combinations of these features as input to binary classification SVM in order to construct incremental kernels. As an example for a kernel, they defined a token kernel as

$$K_T(T_1, T_2) = I(T_1^{word}, T_2^{word}) + I(T_1^{pos}, T_2^{pos}) + I(T_1^{base}, T_2^{base})$$

..., where *base* is the morphological base of a word and I a string comparison function that gives 1 for $I(x, y), x = y$ and 0 otherwise. Important they found entities and head words. Dependency (between the two arguments and more local) and sequence kernels between the two kernels each improved the f-measure by over 7 points, but the dependency kernel over the sequence kernel improved only by nearly two points.

2 Methods

We used the SVMTool(Giménez & Màrquez, 2004)[2] as a POS tagger, BIOs(Mihai Surdeanu & Comelles, 2005)[3] as a Chunker, SwiRL(Màrquez, Surdeanu, Comas, & Turmo, 2005)[4] as a semantic role labeller, and MALT(Nivre, J., Hall, Nilsson, & Marinov, 2006)[5] as a dependency parser in order to first annotate all words in the corpus. Then we extracted combinations of name entities in sentences together with extracted annotations for training of an SVM in YALE(Mierswa, Wurst, Klinkenberg, Scholz, & Euler, 2006)[6] for classifications of targets extracted from the ACE training, searching for a good f-measure on feature selection.

Many preliminary experiments with a configuration of word, POS, chunk, SR, DP, and several bigrams on feature selection gave performances peeking at around 60% with k-nearest neighbor. We tried and compared many runs of feature reduction in YALE with mostly kNN as a wrapper in 5-fold cross-validation. E.g. $k = 5$ resulted in a reduced set of six features that yielded 61% accuracy (with good balance between the different targets) on the type classification task (without subtypes). These six features were extracted from a set of 65 initially chosen features:

$$\left(word_{arg^2}, word_{arg^1}, NE_{arg^1+1}, NE_{arg^2}, SR_{arg^2}, bigram_{m-1}\right)$$

[2]SVMTool: `http://www.lsi.upc.edu/~nlp/SVMTool/`

[3]BIOs: `http://www.lsi.upc.edu/~surdeanu/bios.html`

[4]SwiRL: `http://www.lsi.upc.edu/~surdeanu/swirl.html`

[5]MALT: `http://w3.msi.vxu.se/~nivre/research/MaltParser.html`

[6]Developed at the university of Dortmund, managed by Ingo Mierswa. Project site: `http://sourceforge.net/projects/yale`

This emphasized for us the importance of words, name entities, and more generally the path, and alerted us to bigrams. We increased the number of features in the middle, and introduced bigrams at outer ends of arg_1 and arg_2. Another exemplary variable selection with k-nearest neighbor ($k = 4$) as wrapper in 5-fold cross-validation, yielded 62% accuracy with these seven attributes chosen.

$$\left(word_{arg^2}, NE_{arg^1+1}, word_{arg^1}, NE_{arg^2}, bigram_{arg^2+1}, POS_{arg^1-l}, POS_{arg^2}\right)$$

While it can not be excluded, that classification with other models and/or on other corpora will necessitate very distinct feature sets, we had the growing suspicion, that classification relies chiefly on words and chunking information. We were surprised that now two of seven marginals were included, while the features between the two arguments were comparatively under-represented, and at the prominence of POS. We also estimated that an increase of m and l would not promote performance.

For the rest of our experiments we chose the following feature representation:

Index	**position in text**	**features**
1–5	arg_1	word,POS,chunk,SR,DP
6–10	arg_2	word,POS,chunk,SR,DP
11–22	For three positions to the left of arg_1	POS,chunk,SR,DP
23–24	left of arg_1	bigrams
25–64	between arg_1 and arg_2 8 positions	word,POS,chunk,SR,DP
65–76	right of arg_2, 3 positions	POS,chunk,SR,DP
77–78	right of arg_2	bigrams
79–85	between arg_1 and arg_2	bigrams
86	between arg_1 and arg_2	first bigram
87	between arg_1 and arg_2	second bigram

We considered using WordNet(Fellbaum, 1998) based Jiang and Conrath similarity(J. & D., 1997)[7] of nouns to different types/subtypes (as suggested by (Budanitsky, 2001)), however considerations of time dicouraged us (calculating the similarity between a single pair of words took 11 seconds on a 1.2GHz Power PC G4 with 768 MB RAM running MacOS X). We tried out using hypernym information but as this seemed to complicate the task rather than make it easier we left it out in following analyses.

Continuing with preliminary experiments, we reached 36.78% in a Multilayer Perceptron, 5-fold cross-validation. Starting with a feature set for which we had indication that it was promising, we reduced the data set with a k-nn ($k = 2$) $6, 1, 8, 27, 7$, which yielded 57.40%, $k = 2$ accuracy. on all data (5-fold cross-validation). A k-NN (k=5) reached 57.17% with features 6 and 1. Initial 47,38% k=1, kNN were improved to 60.65% (k=1) in a kNN comparing only three features $(6, 1, 27)$.

[7] `http://search.cpan.org/~sid/WordNet-Similarity/`

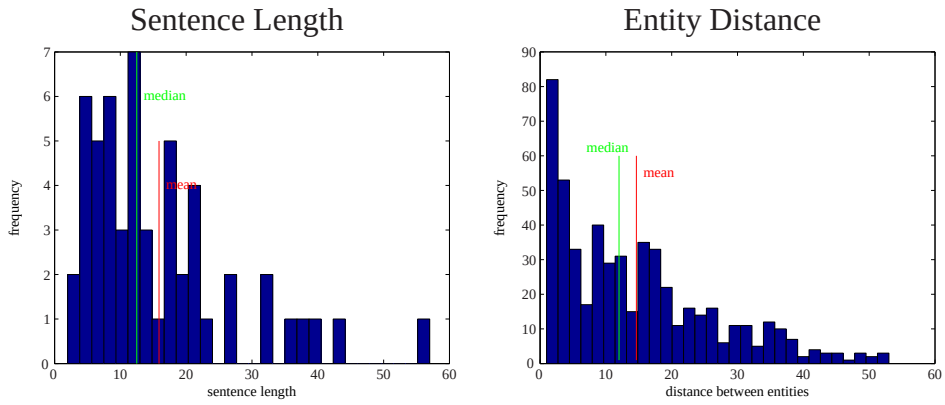


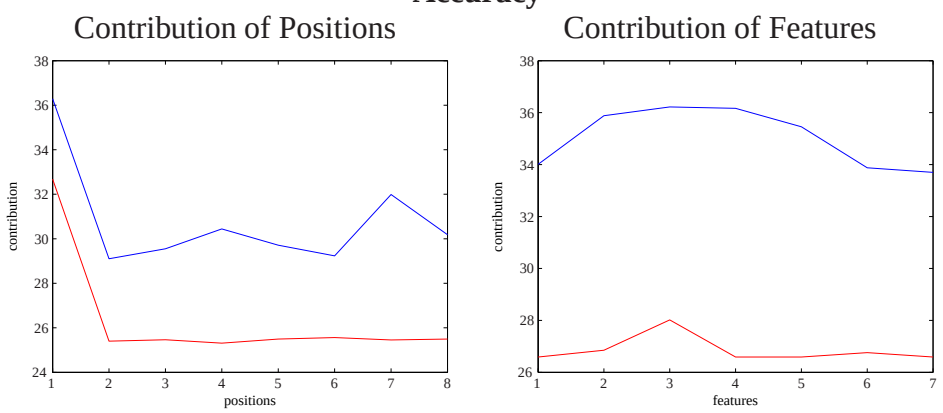
Figure 1: *Histograms of Distances in Sample File of ACE Training Set*

Left: Frequency histogram of sentence lengths of a sample file (mean 15.87, median 12.5) indicates a short window of features could be successful. **Right:** We also compared distances between name entity mentions in the ACL files (mean 14.63, median 12) and found the number of words in the middle to be sufficient to include all words between arguments for more than 32% of between-entity distances. Note that these distances are between all name entities within sentences and not necessarily arguments for relations (and may differ between documents and document categories). We found distances between arguments generally shorter.

We split the data set into fifteen sets with binary classes in a *one-against-one* approach, calculated different models, and combined scores from sets in a *MaxWins* strategy similar to (J. & D., 1997). It is chosen the class with the highest score, in a tie situation, when two or more classes have equal scores, the scoring against the other classes in the tie was compared, if that did not help, the class in the tie with the highest observed frequency was chosen.

We attempted to evaluate the contribution of positions by 5-fold cross-validation

Relative Importance of Positions and Features in Terms of F-measure and Accuracy



The plots can give impressions of contributions of positions and features, respectively, in terms of (**red**) accuracy and (**blue**) f-measure. Positions and feature types were taken in eight and seven groups (see representation above) and sets generated that were stripped of one of a group at a turn and tested in 5-fold cross-validation. The contribution of positions and features is expressed as the inverse of the performance without them. The ordinate gives means of JMySVM running with 5-fold cross-validation over the binary class files. For a meaningful interpretation, attention has to be taken with respect to the different sizes of the sets. Also, features and positions are not independent of each other.

Standard deviations were generally high. As for features, overall, absolute differences between features were minimal but performances expressed an increased influence (in ascending order) of bigrams (total), DP, POS, SR, chunks. The relative importance of the bigrams after arg_2 shows from both plots.

3 Evaluation

We extensively tested a myriad of parameters for SVM, such as kernel functions, c, and different programs, in different feature combinations with results around 17% and peaking at 20% accuracy levels, until a custom kernel working with SVM light[8](?, ?), that combined arguments and bigrams did miraculous work and catapulted us to around 70% accuracy and 70% f-measure.

We began experimenting with custom kernels (and $c = 1$) using an argument kernel (K_{arg}) and reached 70.4% accuracy, 68.3% f-measure. Adding bigrams on left, right, and first and last bigrams in the middle, 70.9% accuracy, 70.47% precision 68.17% recall, 69% f-measure was yielded. We reduced the bigram kernel to first and last in the middle (K_b) and reached 71.7% accuracy, 70% f-measure,

[8]http://svmlight.joachims.org/

precision 71.3%, and recall 60.6%. A quadratic combination of argument and bigram kernel (reduced in the following) gave 70.67% accuracy, 70.44% precision, 67.09% recall, 69.14% f-measure. The cubic combination resulted in 69.19%, 65.24% precision, 62.16% recall.

We added K_{link}, where *link* refers to words and POS tags between arg_1 and arg_2, and with a linear kernel combination of K_{arg}, K_b, and K_{link} we reached 68.78% accuracy, 67.2% precision, 65.35% recall, and 66.27% f-measure. Adding chunking information to K_{link} did not show improved results (68.52% accuracy, 66.47% precision, 65.37% recall, 65.9% f-measure). Omitting K_b (only K_{arg} and K_{link}) gave 69.25% accuracy, 67.09% precision, 65.54% recall, 66.3% f-measure. Omitting K_{arg} showed that K_{arg} is most important of the three (followed by K_{link} as per previous experiment) with 41.60% accuracy, 36.00% precision, 32.03% recall, and 34.25% f-measure.

Feeling experimental, we tried $\left(K_{arg} + K_{link}\right) + \left(K_{arg} + K_{link}\right)^2 + K_b$ which gave 68.94% accuracy, 67.93% precision, 65.31% recall, 66.59% f-measure. All bigrams in the middle for K_b yielded 68.41%, 66.88%, 64.69%, and 65.76%. Equally with all bigrams in the middle the next two experiments $\left(K_{arg} + K_{link}\right)^2 + \frac{1}{3}\left(K_{arg} + K_b\right)^2$ gave 68.31%, 67.78%, 64.62%, 66.16% and $\left(K_{arg} + K_b\right)$ gave 70.00%, 68.88%, 66.74%, 67.82%, which did not convince us of the need to introduce more bigrams.

Class specific results in best configuration:

	Precision	Recall	F-Measure
PART-WHOLE	0.6901	0.662337662	0.675933884
PHYS	0.666044776	0.759574468	0.709741551
ORG-ARFF	0.773480663	0.826771654	0.79923882
GEN-AFF	0.630139073	0.454545455	0.528129321
ART	0.726744186	0.618811881	0.668449198
PER-SOC	0.798076923	0.794258373	0.79616307
Total	0.714097604	0.686049915	0.7

4 Conclusions

We studied the classification of relationships from ACE documents in support vector machine (SVM). Features tried out were the actual words, bigrams of words, annotations, such as part of speech (POS), semantic roles (SR), dependency parses (DP), and NEs. Of the different kernels, custom kernels, similar to (Zhao & Grishman, 2005), were superior by a huge margin to all other, standard kernels.

This classification task is non-trivial. We did not include syntax trees, nor did

we have base words other than of verbs. We think that our performance could be improved significantly with WordNet information as mentioned before. A sampling of predictions in combinations of feature types could provide to our understanding of these and possibly help us give better results. We did not try any ngrams with n higher than 2 (bigrams), so that might also be a way of increasing classification performance. We think that increasing the number of words in the kernels does not necessarily have an increase of performance as a consequence.

References

Budanitsky, A. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. *Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*.

Fellbaum, C. (Ed.). (1998). *Wordnet: an electronic lexical database (language, speech, and communication)*. Cambridge, Massachusetts (USA): MIT Press.

Giménez, J., & Màrquez, L. (2004). Svmtool: A general pos tagger generator based on support vector machines. *Proceedings of the 4th LREC*.

J., J., & D., C. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of International Conference on Research in Computational Linguistics. Taiwan*.

Màrquez, L., Surdeanu, M., Comas, P., & Turmo, J. (2005). A robust combination strategy for semantic role labeling. *Proceedings of HLT/EMNLP*.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*.

Mihai Surdeanu, J. T., & Comelles, E. (2005, September). Named entity recognition from spontaneous open-domain speech. *Proceedings of the 9th International Conference on Speech Communication and Technology (Interspeech)*.

NIST. (2006). *The ace06 evaluation plan.* ([Online; accessed 20-January-2007])

Nivre, J., Hall, J., Nilsson, G., J. and Eryigit, & Marinov, S. (2006). Labeled pseudo-projective dependency parsing with support vector machines. *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*.

Zhao, S., & Grishman, R. (2005, June). Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd annual meeting of*

the association for computational linguistics (acl'05) (pp. 419–426). Ann Arbor, Michigan: Association for Computational Linguistics.

Zhou, G., Su, J., Zhang, J., & Zhang, M. (2005, June). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)* (pp. 427–434). Ann Arbor, Michigan: Association for Computational Linguistics.