

An effective method of using Web based information for Relation Extraction

Yong Wai Keong, Stanley

Institute for Inforcomm Research
21 Heng Mui Keng Terrace,
Singapore 119613
wkyong@i2r.a-star.edu.sg

Su Jian

Institute for Inforcomm Research
21 Heng Mui Keng Terrace,
Singapore 119613
sujian@i2r.a-star.edu.sg

Abstract

We propose a method that incorporates paraphrase information from the Web to boost the performance of a supervised relation extraction system. Contextual information is extracted from the Web using a semi-supervised process, and summarized by skip-bigram overlap measures over the entire extract. This allows the capture of local contextual information as well as more distant associations. We observe a statistically significant boost in relation extraction performance.

We investigate two extensions, thematic clustering and hypernym expansion. In tandem with thematic clustering to reduce noise in our paraphrase extraction, we attempt to increase the coverage of our search for paraphrases using hypernym expansion.

Evaluation of our method on the ACE 2004 corpus shows that it out-performs the baseline SVM-based supervised learning algorithm across almost all major ACE relation types, by a margin of up to 31%.

1 Introduction and motivation

In this paper, we shall be primarily dealing with the sort of relations defined in the NIST's Automatic Content Extraction program, specifically for the Relation Detection and Characterization (RDC) task (Dodgington et al., 2004). These are links between two entities mentioned in the same sentence,

and further restrict our consideration to those relationships clearly supported by evidence in the scope of the same document.

The ACE's annotators mark all mentions of relations where there is a direct syntactic connection between the entities, i.e. when one entity mention modifies another one, or when two entity mentions are arguments of the same event. Relations between entities that are implied in the text but which do not satisfy either requirement are considered to be implicit, and are marked only once.

Our work sits squarely in the realm of work on regular IE done by (Zelenko et al., 2003; Zhou et al., 2005; Chen et al., 2006). Here, the corpus of interest is a well defined set of texts, such as news articles, and we have to detect and classify all appearances of relations from a set of given relation types in the documents. In line with assumptions in the related work, we assert that the differences in the markup for implicit and explicit relations does not significantly affect our performance.

Supervised learning methods have proved to be some of the most effective for regular IE. They do however, need large volumes of expensively annotated examples to perform robustly. As a result, even the large ACE compilation has deficiencies in the number of instances available for some of the relation types. (Zhang et al., 2005) reports an F-score of 50% for categories of intermediate size and only 30% for the least common relation types. It appears that there are hard limits on the performance of relation extraction systems as long as they have to rely solely on information in the training set.

We were thus inspired to explore how one could

exploit the Web, the largest raw text collection freely available, for regular IE. In this paper, we detail the ways one can fruitfully employ relation specific sentences retrieved from the Web with a semi-supervised labeling approach. Most importantly we show how the output from such an approach can be combined with existing knowledge gleaned from supervised learning to improve the performance of relation extraction significantly.

2 Related Work and Differences

To our knowledge, there is no previous work that exploits the information from a large raw text corpus like the Web to improve supervised relation extraction. In the spirit of the work done by (Shinyama and Sekine, 2003; Bunescu and Mooney, 2007), we are trying to collect clusters of paraphrases for given relation mentions. Briefly, since the same relation can be expressed in many ways, the information we may learn about that relation in any single sentence is very limited. The idea then is to alleviate this bias by collecting many paraphrases of the same relation instance into clusters when we train our system.

Shinyama generalizes the expressions using part-of-speech information and dependency tree similarity into generic templates. Bunescu’s work uses a relation kernel on subsequences of words developed in (Bunescu and Mooney, 2005). We observed that both approaches suffer from low recall despite the attempts to generalize the subsequences and templates probably because they rely on local context only.

Based on our observation, we looked for a way to use our clusters without losing non-local information about the sentences. Bag-of-words or unigram representations of our paraphrase clusters are easy to compute, but information about word ordering is lost. Hence, we settled on the use of a skip-bigram representation of our relation clusters instead.

Skip-bigrams (Lin and Och, 2004) are pairs of words in sentence order allowing for gaps in between. Longer gaps capture far-flung associations between words, while short gaps of between 1 and 4 capture local structure. In using them, we do not restrict ourselves to context centered around the entity mentions. Another advantage of using skip-bigrams is that we can capture some extra-sentential infor-

mation since we are no longer restricted by the ability to generate dependency structures within a single sentence as Shinyama is.

Using skip-bigrams, we can assess the similarity of a particular new relation mention instance against the relation clusters we collect in training. We can then compute a likelihood that we combine with the predictions of the supervised learning algorithm for final classification.

Two possible extensions to the basic method stated above were examined.

A central problem with the paraphrase collection approach when applied to an open corpus is noise. As pointed out by (Bunescu and Mooney, 2007), even though the same entities co-occur in multiple sentences, they are not necessarily linked by the same relationship in all of them. The problem is exacerbated when the open corpus we look at contains documents from heterogenous domains. Indeed, we cannot even assume that the predominant relation that holds between two entities in the set of sentences is the relation of interest to us.

One means of combating this is suggested by (Bunescu and Mooney, 2007). They re-weight the importance of word features in their model to reduce topic drift. We try a different solution based on the thematic clustering of sentences. Sentences extracted from the raw corpus are mapped to a vector space and partitioned into different clusters using the Partitioning Around Medoids algorithm (Kaufmann and Rousseeuw, 1987). Sentences in the clusters closest to the original relation mention instance are more likely to embody the same relationship. Hence we retain such clusters, while discarding the rest. As the relation we wish to recover may not be the predominant one, the cluster that is retained is also often not the largest one.

Another problem identified by Shinyama is that the same entity may itself be referred to in different ways. If the form used in the original relation mention is uncommon, then few paraphrases will be found. For instance, “President Bush” may be referred to as “Dubya” by a writer. Searching for sentences online with the word “Dubya” and the other entity participating in the relation is likely to result in a collection heavily biased towards the originator of the nickname. Shinyama’s solution is to use a limited form of co-reference resolution to re-

place these forms with a more general noun phrase. As co-reference resolution is itself an unreliable process, we suggest the use of hypernym substitution instead.

In subsequent sections, we will outline the structure of our system, examine the experimental evidence of its viability using the ACE program data, and finish with a discussion of the extensions.

3 Overall structure

Our system is organized very naturally into two main phases, a learning or training phase, followed by a usage or testing phase. The learning phase is subdivided into two parallel paths, reflecting the hybrid nature of our algorithm.

Fully supervised learning based on annotations takes place in tandem with a semi-supervised algorithm that captures the paraphrase clusters using entity mention instances. We will combine the models from both the supervised learning and the semi-supervised algorithm using a meta-classifier trained a different subset of the data.

3.1 Learning Procedure

Our goal is to acquire as much contextual information from the available annotations as possible via our supervised learner and expand on that using Web based information found by our semi-supervised algorithm.

We constructed our fully supervised learner according to the specifications for the system developed by (Zhou et al., 2005). It utilizes a feature based framework with a Support Vector Machine (SVM) classifier (Cortes and Vapnik, 1995). We support the same set of features as Zhou, namely: local word position features, entity type, base phrase chunking features, dependency and parse tree features as well as semantic information like country names and a list of trigger words. In our current work, we use Michael Collins' (Collins, 2003) parser for syntactic information.

Sentence boundary detection, chunking, and parsing are done as preprocessing steps before we begin our learning.

Given a sentence with the relation mention instance, the semi-supervised method goes through the following five stages:

1. From the list of entities marked in the sentence, generate all possible pairings as candidates. We pick one of these candidates and proceed to the next step.
2. Gather hypernyms for each entity mention using Wordnet synsets and generate all possible combinations of entity pairs from the two sets.
3. Find sentences from the Web that mention both entities.
4. Cluster the sentences found using k-medoids, filter out noise and retain only the cluster that the original relation mention would be assigned to.
5. Collate all clusters by relation type and generate a skip-bigram index for each relation type.

We will spend the rest of this section on the details of each stage.

3.1.1 Extracting entity mentions and gathering hypernyms

The process starts when we receive a relation mention and the sentence it was found in, for instance in the following sentence from the ACE 2004 corpus: "As president of Sotheby's, she often conducted the biggest, the highest profile auctions."

From the annotation, we find that the two entities of interest are *president* and *Sotheby's* in the *EMP-ORG Executive* relation, which we could represent as the predicate *EMP – ORG Executive(em₁, em₂)*. Since our aim is to find as many instances of semantically similar sentences as possible, we want to lessen the negative impact of quirky spelling or naming in a given sentence. Hence we do a hypernym search in Wordnet to find more general terms for our entities and create list of similarly related entities (*em₁'t*, *em₂'t*) etc. If the mention is a named entity, we do without the hypernym expansion and use coreference information (when available) to find the most common substring amongst the mentions for the same entity.

In this example, it might result in the four pairs show in Table 1.

Table 1: Examples of entity pairs after hypernym expansion

President	Sotheby's
Chief Executive	Sotheby's
Decision maker	Sotheby's
leader	Sotheby's

Table 2: Examples of extracted text from Google

The 40 year old former president travels incognito to Sotheby's
Brooks was named president of Sotheby's Inc.
Subsidiary President at Sotheby's...
Bill Ruprecht, chief executive of Sotheby's, agreed that September 11 had been...
The Duke will also remain leader of Sotheby's Germany...

3.1.2 Web search

(Geleijnse and Korst, 2006) use Google as their search engine for extracting surface patterns from Web documents. We use the same procedure here to find our paraphrases. For each pair of arguments, we create a boolean query string $em_1 * em_2$, and submit it to Google. The query will find documents where mentions of em_1 are separated from em_2 by any number of words. We restrict the language to English.

Google returns a two line extract from the documents that match our boolean query. The extracts are generally those lines where the key query terms are most densely collocated. These are parsed and obviously nonsensical sentences are discarded based on the occurrence of words from a list of stop words. If a group of sentences are very similar, we choose a single representative and discard the rest. For every remaining sentence, we normalize them by removing extraneous HTML tags. Some examples of extracts found are listed in Table 2.

3.1.3 Cluster, filter and collate

In general, the collection of sentence extracts we have at the end of the previous stage are likely to be about a diverse range of topics. As we are only interested in the subset that is most compatible with the

thematic content of our original relation mention, we will have to filter away unrelated extracts. For example, the *EMP-ORG Executive* relation does not hold in the sentence: "The 40 year old former president travels incognito to Sotheby's".

We make use of the K-medoids method by (Kaufmann and Rousseeuw, 1987). The terms from all sentences are extracted and their frequency in each sentence is computed. Each sentence is now mapped as a vector of frequencies in the space of the terms that we observed. The resulting vectors are stored as a large matrix. Picking a random partition of the sentences as our starting point, we assign some sentences to be the cluster centers, and iteratively refine the clusters based on a distance minimization heuristic.

Through some preliminary experiments, we find that K-medoids based clustering with 5 classes produced the most consistent results. From a list of excerpts, our algorithm culls the sentences that belonged to the 4 irrelevant clusters and produces the excerpts which capture the original relationship best. Since the quality of the partitions produced by the algorithm is sensitive to the initial random start, we do this process twice with different configurations and take the union of the two clusters as our final result.

The best excerpts are stored with accompanying meta-data about the originating training relation instance in what we call *pseudo-documents*. We group the pseudo-documents in our database by the relation label that the instance pairs were given. Thus we end up with several bags of pseudo-documents, where each bag corresponds to a single relation type of interest.

For computational efficiency, we generate an inverted hash-index for the pseudo-documents. Our skip-bigrams act as the keys and the records are lists of meta-data nodes. Each node records the sentence that the bigram is observed in, the relation type of that sentence, and the position of the bigram.

All we need now is a means of measuring the similarity of a new relation mention instance with the bags of pseudo-documents to assign a relation label to it.

3.1.4 Skip-bigrams

As discussed in the introduction, instead of generalizing our bags of documents into patterns or relation extraction kernels, we create skip-bigram indices. There are several advantages in doing so.

Skip-bigrams are easy to compute relative to the dependency trees or subsequence kernels used in (Shinyama and Sekine, 2003) or (Bunescu and Mooney, 2005). Moreover, we can tune the number of gaps allowed to capture long-distance word dependency information, which is not done by the other approaches because it is relatively more expensive for them to do so, due to the combinatorial explosion. In addition, as compared to Bunescu’s subsequence approach which needs 4 words to match, bigrams are far less likely to be sparse in the document space.

Since we relied upon skip-bigrams in our queries to Google, it is only natural that we use it again in assessing the similarity of two pseudo-documents. Each pseudo-document is really an extractive summary of online articles about the same topic, with the same entities. The degree of overlap between two pseudo-documents is a good measure of their thematic overlap.

Now that we have a metric, that still leaves the question of our matching heuristic open. Do we automatically assign a test instance the relation label of the sentence with the highest skip-bigram overlap? This naive approach is problematic. In general, longer sentences will have more bigrams and hence higher probability of overlapping with other sentences. We could normalize the bigram overlap score by the length of the sentences, but here we leave the optimization to a machine learner.

Another possible heuristic is to pick the relation label whose bag has the highest number of matching bigrams with our test instance. Again, this will be biased, but now towards bags with larger numbers of pseudo-documents. A last possibility is to look at the total number of sentences that have bigram matches, and weight the overlap score higher for those with more sentence matches.

Therefore, instead of designing the heuristic explicitly, we use a validation set to observe the statistical correlations of each of the three possible heuristics we discussed above. We train an additional

model, using an SVM to choose the weights for each heuristic automatically.

Accordingly, we do the following for each validation instance, V , and its pseudo-document P .

For each extracted sentence j in pseudo-document P , we look up the database of pseudo-documents from our training set, and compute the skip-bigram similarity with every single sentence. We have a skip-bigram similarity score for every single sentence in the database with respect to V . The scores are collated according to the relation classes.

For each relation class we generate three numbers, *TopS*, *Matching docs*, and *Total*. Using the notation by (Lin and Och, 2004), we denote the skip-bigram overlap between two sentences X and Y as $Skip2(X, Y)$. For the i^{th} relation, C_i is the set of all pseudo-documents in our training set of that relation type.

$$TopS = \max_{Y \in C_i, j \in P} Skip2(P_j, Y) \quad (1)$$

$$Matching = \sum_{Y \in C_i} \sum_{j \in P} I[Skip2(P_j, Y) > 0] \quad (2)$$

$$Total = \sum_{Y \in C_i} \sum_{j \in P} Skip2(P_j, Y) \quad (3)$$

The three figures provide a summary of the best sentence level match, and the overall relation level overlap in terms of the number of sentences and number of overlaps. As an illustration, we consider the case where we have two sentences in our pseudo-document $P = P_1, P_2$ and a relation RX . We compute $Skip2(P_1, Y)$ and $Skip2(P_2, Y)$ by looking up the skip-bigrams in the database for RX and aggregating over sentences. Let’s assume that $|RX| = 3$ and only $Skip2(P_1, Y_1) = 2$, $Skip2(P_1, Y_3) = 5$, $Skip2(P_2, Y_3) = 4$ are non-zero. Then *TopS* for instance V is 5. *Matching* will be 2 since only Y_1 and Y_3 have overlaps with elements of P . The *Total* is simply $2 + 5 + 4 = 11$.

3.2 Combining supervised with semi-supervised models

After the preceding steps, we have a trained SVM model based, and our skip-bigram index from the semi-supervised procedure. In this section, we will describe a method of combining these into a better classifier.

The validation data we left aside earlier is sent through our system with the relation labels removed. Each entity pair in this validation set has a corresponding pseudo-document and a file with numerical features for the SVM model.

An instance V is scored by Zhou’s SVM classifier, which assigns a relation tag, V_{ST} to it. In parallel, the skip-bigram assessment results in $\{TopS, Matching, Total\}$ scores for each of the relation classes. We treat the tag and numbers as features for training another SVM, which we shall refer to as SVM_C . This is our final meta-classifier for relation extraction on the tenth of the original data set aside for testing. The meta-classifier may also be used for completely new data.

4 Experimentation

We use the ACE corpus provided by the LDC from 2004 to train and evaluate our system. There are 674 annotated text documents and 9683 relation instances in the set.

Starting with a single training set provided by the user, we split that into three parts: the majority (80%) is used for the learning phase, one tenth is used for the validation during construction of the combined model, and the remaining tenth is used for testing. We ran a series of experiments, using five-fold cross-validation.

Unlike typical cross-validation, the fifth that we set aside is further sub-divided into two parts as we stated before. Half is used when we construct the hybrid model merging supervised and semi-supervised paths, and the remainder is used for the actual testing and evaluation.

We used the 6 main relation types defined in the annotation for the ACE dataset: ART, EMP-ORG, GPE-AFF, OTHER-AFF, PER-SOC, and PHYS. We computed three measures of the effectiveness, the recall (R), precision (P) and F-1 score (F-1).

4.1 Comparison against the baseline

The first set of experiments we shall discuss compares the overall system against our baseline. The baseline system is implemented as a feature extraction module on top of a set of binary SVM classifiers. A primary classifier is used to separate the candidates without any relation of interest from the

rest. Secondary classifiers for each relation type are then used to partition the positive candidates by voting. The performance of our baseline classifier when tested on the ACE2003 dataset is statistically indistinguishable from that reported by Zhou et al. in (Zhou et al., 2005).

Drilling down to the level of individual relation classes as shown below, we note that the meta-classifier performs better than the baseline on all but one of the relations. This might be due to the inherent ambiguity of the OTHER-AFF class.

Relation	Ratio	System	R	P	F-1
ART	5.6	Hybrid	0.48	0.73	0.59
		baseline	0.29	0.43	0.34
EMP-ORG	40.0	Hybrid	0.78	0.75	0.76
		baseline	0.67	0.83	0.73
GPE-AFF	11.7	Hybrid	0.49	0.59	0.53
		baseline	0.36	0.56	0.45
OTHER-AFF	3.4	Hybrid	0.14	0.18	0.16
		baseline	0.18	0.59	0.28
PER-SOC	9.7	Hybrid	0.63	0.80	0.70
		baseline	0.32	0.5	0.39
PHYS	29.4	Hybrid	0.75	0.59	0.66
		baseline	0.40	0.64	0.49

The hybrid system has slightly lower precision on the two largest relation classes, EMP-ORG and PHYS, but higher recall, resulting in better F-scores on both types. Finally, note that on the three intermediate sized classes, ART, PER-SOC, and GPE-AFF, the recall and precision were both higher. The results suggest that the Web information does improve recall substantially, but affects precision in cases where there already is a substantial amount of training data. It confirms our original assertion that the hybrid approach works well for mid-sized relation classes, where the amount of training data is not enough for the supervised system to perform optimally.

We use the T-test to see if our improvements over the baseline were significant at a 0.05 level. To summarize, recall for the PHYS, PER-SOC, GPE-AFF and EMP-ORG relations was improved significantly. The difference in precision is significant for the PER-SOC, and OTHER-AFF classes, while F-1 score differences for the PER-SOC and PHYS relation classes at 0.00085, 0.032 respectively were both significant.

4.2 Testing hypernym expansion and clustering

Subsequent experiments were aimed at quantifying the contribution of hypernym expansion and thematic clustering to our hybrid system. We ran a 2 factorial experiment with four of our five folds, where we take the hypernym expansion and clustering as treatments. Since we have more than one feature being tested, and we wish to observe the relative contribution of each factor, we used an ANOVA test instead of T-tests (Montgomery, 2005).

Our intuitive justification for hypernym expansion was that recall would be boosted for relation types where the name entities tend to be overly specific in the training corpus. Place names, personal names and the object names are obvious targets. Indeed, we noted that the recall did increase in absolute terms on average for the ART, PER-SOC and PHYS relation types (about 3% for each), but declined slightly (about 0.5%) for the rest. Overall however, the size of the effects was too small to be statistically significant. This suggests that other methods of term generalization may be needed to achieve a larger effect.

Next, we looked at the contribution of clustering. Our initial experiments showed that k-medoids with 5 clusters was able to produce very precise clusters. However, it would be at the expense of some of the potential gain in recall from Web extracts.

Our experiments shows that clustering does indeed lower the potential recall. However, the hoped for improvement in precision was observed only in the PER-SOC (6%) and GPE-AFF (0.7%) relations. This suggests that the effect of name entities having multiple relations is concentrated in the classes of named entities related to Persons and GPEs. Again, the size of the effects was not statistically significant.

A more thorough investigation of clustering techniques with different settings for k and different algorithms will be needed before we can make stronger statements.

5 Discussion and Conclusion

We have presented a hybrid approach to relation extraction that incorporates Web based information successfully to boost the performance of state-of-the-art supervised feature based systems. Evaluation on the ACE corpus shows that our skip-bigram based

relevance measures for finding the right paraphrase in our Web extract database are very effective.

While our analysis shows that the addition of clustering and hypernym expansion to the skip-bigram based process is not statistically significant, we have indications that the effect on recall and precision is positive for certain relation classes.

In future work, we will examine improvements to the clustering algorithm to reduce the impact on recall. We will look at alternative ways of attacking the problem of name entity generalization and assess the impact of methods like co-reference resolution in the same ANOVA framework.

Acknowledgment

This research is supported by a Specific Targeted Research Project (STREP) of the European Union's 6th Framework Programme within IST call 4, Bootstrapping Of Ontologies and Terminologies STtrategic REsearch Project (BOOTStrep).

References

- Razvan C. Bunescu and Raymond J. Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS*.
- Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, Prague, Czech Republic, June. Association for Computational Linguistics.
- Jinxu Chen, Donghong Ji, Chew Lim Tan, and Zheng-Yu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *ACL*. The Association for Computer Linguistics.
- Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. volume 20, pages 273–297.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program. tasks, data and evaluation.
- G. Geleijnse and J. Korst. 2006. Learning effective surface text patterns for information extraction. In *Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 1 – 8, Trento, Italy,

- April. The Association for Computational Linguistics.
http://acl.ldc.upenn.edu/eacl2006/ws06_atem.pdf.
- L. Kaufmann and P. J. Rousseeuw. 1987. Clustering by means of medoids. In Y. Dodge, editor, *Statistical Data Analysis based on the L1 Norm*, pages 405–416, Amsterdam. Elsevier/North Holland.
- Chin Y. Lin and Franz J. Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of Coling 2004*, pages 501–507, Geneva, Switzerland, Aug FebruaryMarch–Aug FebruaryJuly. COLING.
- Douglas C. Montgomery. 2005. *Design and analysis of experiments*. John Wiley And Sons.
- Yusuke Shinyama and Satoshi Sekine. 2003. Paraphrase acquisition for information extraction. In Kentaro Inui and Ulf Hermjakob, editors, *Proceedings of the Second International Workshop on Paraphrasing*, pages 65–71.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.
- Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *IJCNLP*, pages 378–389.
- Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*. The Association for Computer Linguistics.