Does Google own Youtube?
Entity Relationship Extraction with Minimal Supervision

Keywords: semi-supervised learning, multiple instance learning, entity relation extraction, SVM, WLS-SVM

Jan De Belder JAN.DEBELDER@STUDENT.KULEUVEN.BE
Wim De Smet WIM.DESMET@CS.KULEUVEN.BE
Raquel Mochales RAQUEL.MOCHALES@CS.KULEUVEN.BE
Marie-Francine Moens MARIE-FRANCINE.MOENS@CS.KULEUVEN.BE
Department of Computer Science, K.U.Leuven, Celestijnenlaan 200A, Leuven, Belgium

Abstract

This paper advances state of the art entity relationship extraction from text with minimal supervision in two ways. The contributions are a new weighting scheme, that solves different types of bias caused by having only a few training examples, and a new method to construct a support vector machine in the Multiple Instance Learning (MIL) setting, based on Weighted Least Squares (WLS).

1. Introduction

Entity Relationship Extraction (ERE) from text is a problem in the domain of information extraction, and is receiving growing attention. The aim is to identify relationships between two entities in texts, such as *BornIn*(Person,Location). Supervised Learning can solve this problem, but has its obvious drawback of manual labor.

This paper builds on the research performed by (Bunescu & Mooney, 2007), where a new and promising method for ERE with minimal supervision is introduced. Its main idea is to supply only a few pairs of entities, of whom it is well known that they do or do not express a particular relationship, e.g. *Acquired*(Google,YouTube) and ¬*Acquired*(Apple,Google). For every pair, a set of sentences containing both entities is downloaded from the Word Wide Web. Not all sentences for the positive pairs will contain the wanted relationship, but quite a few of them will. For the negative pairs, it is safe to assume that none of the sentences will express the relation. This reduces the problem to a Multiple Instance Learning (MIL) problem. MIL problems are defined by positive and negative bags (i.c. a bag of sentences for every pair). The negative bags contain absolutely no positive examples, whereas every positive bags contains at least one positive example.

This paper is structured as follows. First, section 2 summarizes the background work presented in (Bunescu & Mooney, 2007). Then, section 3 introduces Weighted Least Squares SVM, as an alternative to the Quadratic Programming (QP) formulation of MIL for SVMs in previous work. In section 4, a solution for the two types of bias encountered in (Bunescu & Mooney, 2007) is proposed, and a new type is introduced. The results can be found in section 5, followed by a summary in section 6.

+/-	Arg e_1	Arg e_2	Size
+	Google	YouTube	1742
+	Adobe Systems	Macromedia	1301
+	Viacom	DreamWorks	792
+	Novartis	Eon Labs	739
-	Yahoo	Microsoft	2901
-	Pfizer	Teva	328
+	Pfizer	Rinat Neuroscience	698 (492)
+	Yahoo	Inktomi	853 (237)
-	Google	Apple	1131
-	Viacom	NBC	528

Table 1. Corporate Acquisition pairs.

2. Method

This section gives an overview of the two key aspects of the method described in (Bunescu & Mooney, 2007): the acquisition of the data, and how to learn from it. The rest of the paper focuses on the Corporate Acquisition relation, with the same pairs used in the

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

previous work. These pairs can be found in table 1. The upper part is used for training, and the sentences found for bottom pairs are used as a test set, with the number of positive sentences between parentheses.

For every pair of entities, e_1 and e_2, a query ”e_1 * * * * * * * e_2” is sent to a search engine (Google). This query is a close approximation of the desired result: web pages with the two entities in the same sentence. The resulting web pages are downloaded, the sentences are extracted, and only those containing both entities are kept. The information in the sentences can be further elaborated by applying POS tagging etc.

2.1. Learning

Despite the fact of having a MIL problem, a somewhat adapted SVM is used. In (Ray & Craven, 2005) it was made clear that Supervised Learning algorithms perform about equally well on MIL problems. The SVM's objective function is modified as follows:

minimize:

$$\frac{1}{2}\|w\|^2 + \frac{C}{L}(c_p \frac{L_n}{L} \xi_p + c_n \frac{L_p}{L} \xi_n)$$

subject to:

$$w\phi(x) + b \geq 1 - \xi_x, \forall x \in \textit{positive bags}$$
$$w\phi(x) + b \leq -1 + \xi_x, \forall x \in \textit{negative bags}$$
$$\xi_x \geq 0,$$

where ξ_p and ξ_n stand for the sum of the slack variables belonging to sentences in positive and negative bags respectively, and the capacity control parameter C is normalized by the total number of sentences ($L = L_p + L_n$). Parameter $c_p = (1 - c_n)$ controls the penalisation of false negatives vs. false positives, the latter being far more severe due to the MIL setting.

The function ϕ maps every sentence into a feature space with a dimension for every pattern of n (≤ 4) words. Mapping every sentence into this space is computationally infeasible, which is why the optimization problem is solved in the dual space. The dot product between two sentences in this space can be efficiently calculated with the String Subsequence Kernel (SKK) (Bunescu & Mooney, 2006). Previous work also used POS tags as patterns. This was excluded here to keep everything language independent.

3. WLS-SVM

Weighted Least Squares SVMs (WLS-SVMs) were introduced by (Suykens et al., 2002), and intended as a more robust alternative to Least Squares SVMs (LS-SVM). LS-SVM is an SVM version which involves equality instead of inequality constraints and works with a least squares cost function. In this way, the solution follows from a linear system instead of a Quadratic Programming problem. However, the estimation of the support values is only optimal in the case of a Gaussian distribution of the error variables. With WLS-SVM robust estimates are obtained by weighting the error variables e_k with a factor v_k, that is based on the distribution of e_k. The optimization problem becomes:

minimize:

$$\frac{1}{2}\|w\|^2 + \frac{1}{2}C \sum v_k e_k^2$$

subject to:

$$w\phi(x_k) + b = y_k - e_k, \forall k,$$

with C as before, and y_k the label. The weights v_k are determined by first training a LS-SVM, and calculating the outliers by means of robust statistics. The outliers are given a lower weight, and the system is retrained.

To adapt this to a MIL setting, we already use a weighted version in the first step: a weight of 0.1 for sentences in positive bags, and 0.9 for those in negative bags. After that, we calculate the error variables e_k the same way, but only reweight the outliers for the positive bags: sentences further away than the standard deviation of e_k are given weight 0.001, with a linear descent starting from 0.8 standard deviations.

4. Three types of bias

As a result of having few bags with many sentences, there are some difficulties that arise. Two types of bias were already defined in (Bunescu & Mooney, 2007):

Type I bias: All sentences in a certain bag contain the two entities, by definition. With any one of these entities, it is possible that they are frequently accompanied by words that are strongly related. For example, YouTube will be correlated with the word ”video”, and Google with the word ”search”. As such, a large number of sentences will contain both of these words, thereby misleading the SVM: if a new negative sentence contains these correlated words, it will very likely be misclassified as expressing the relationship.

(Bunescu & Mooney, 2007) mentioned a solution for type I bias. A weight is given to each word, depending on how much it is correlated to either one of the entities: the more it occurs in sentences with the entity (acquired the same way as in section 2, but for only one entity), the lower the weight. This kernel can be found in the results as SSK-T1. However, despite better results, it is not a perfect approach. A situation where it might work to a disadvantage, is when one of the entities is relatively unknown, and the only media attention it gets is from being involved in the relation.

Type II bias: This type of bias is on the level of a specific instantiation of the relation. For example, the time argument will frequently occur. New negative sentences that mention the same date will look like sentences from that bag, and could be misclassified.

We developed a new weighting scheme for type I and II bias, incorporating the MIL setting at word level. If a word occurs in all bags, it probably helps in expressing the relationship. If a word occurs in only one positive bag, it is possible that it is due to one of the biases mentioned above. This leads to following equation:

$$weight(word) = \frac{\#\, positive\ bags\ that\ contain\ word}{\#\, positive\ bags} \quad (1)$$

Many small improvements are possible for this weighting scheme, including stemming, and stating that a bag contains a word only if it occurs significantly, i.e. more than would be expected from the frequency for that word based on a sufficiently large corpus. More improvement is possible by incorporating the number of negative bags the word is in.

Type III bias (Temporal bias): This new type of bias comes down to the fact that there may be different temporal phases that two entities can be in, prior or after actually being in the wanted relationship. A typical example in this case would be a company making an offer to acquire another company. Sentences expressing this intent can occur for every positive pair, thus making it very difficult to discriminate this relation from the relation wanted. A possible solution is supplying negative examples that express each of the phases that are not wanted. The disadvantage is that this requires more supervision.

5. Results

We tested five kernels, including two new ones, on the Corporate Acquisition relation (table 1). SSK is the standard String Subsequence kernel, without weights. BOW is a simple bag of words kernel, counting the number of common words in two sentences. SSK-T1 gives weights to the words, as defined in (Bunescu & Mooney, 2007) to solve type I bias. The new kernels, BOW-B and SSK-B, both use the weights from equation 1, with a bag of words approach and the String Subsequence Kernel respectivly. Tests were done with the WLS and the QP method. The results consist of the area under the precision-recall curve (obtained by varying the threshold term b), and can be found in table 2. The WLS method gives competitive results when compared to the QP method, with a significant increase for kernels where the new weighting scheme is used. The combination of SSK-B and WLS clearly gives the best result. Inspection of the weights showed that the most important words, like conjugations of *acquiring* and *buying*, receive a higher weight than less relevant words, which explains the results.

An observation worth mentioning is the possible consequence of the choice in training pairs: while manually labelling data, some bags had 10% of the sentences expressing the relation, whereas others had an accuracy of up to 90%.

	SSK	BOW	SSK-T1	BOW-B	SSK-B
WLS:	54.25	22.66	75.01	39.30	**87.88**
QP:	50.27	23.00	66.10	31.16	74.59

Table 2. Area under Precision-Recall curve.

6. Summary

We succesfully improved a state of the art method for entity relation extraction from text with minimal supervision, trained on a large corpus (i.e. the World Wide Web). Experiments show an increase in performance by using a modified WLS-SVM and a new weighting scheme. Both aim at taking the MIL setting into account, the former on the scale of the individual sentences, the latter by doing so at word level. The new weighting scheme resolved several types of previously encountered bias.

Numerous paths are valuable to further investigate, such as alternative weighting schemes and other ways to deal with the noise in the positive bags. Overall, the method is capable of identifying the learned relationship with high accuracy, but it is important to notice that there could be certain relations which are more difficult to learn, depending on the different phases in time the relation goes through, and the influence the choice of entity pairs can have.

References

Bunescu, R., & Mooney, R. (2006). Subsequence kernels for relation extraction. *Advances in Neural Information Processing Systems*, *18*, 171.

Bunescu, R., & Mooney, R. (2007). Learning to extract relations from the web using minimal supervision. *Annual meeting-association for Computational Linguistics* (p. 576).

Ray, S., & Craven, M. (2005). Supervised versus multiple instance learning: an empirical comparison. *Proceedings of the 22nd international conference on Machine learning* (pp. 697–704).

Suykens, J., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, *48*, 85–105.