Procesamiento del Lenguaje Natural, Revista nº 51, septiembre de 2013, pp 41-48 recibido 30-04-2013 revisado 18-06-2013 aceptado 21-06-2013

Removing Noisy Mentions for Distant Supervision

Eliminando Menciones Ruidosas para la Supervisión a Distancia

Ander Intxaurrondo*, Mihai Surdeanu, Oier Lopez de Lacalle***, Eneko Agirre***
*University of the Basque Country. Donostia, 20018, Basque Country
**University of Arizona. Tucson, AZ 85721, USA
***University of Edinburgh. Edinburgh, EH8 9LE, UK
ander.intxaurrondo@ehu.es, msurdeanu@email.arizona.edu,
oier.lopezdelacalle@ehu.es, e.agirre@ehu.es

Resumen: Los métodos para Extracción de Información basados en la Supervisión a Distancia se basan en usar tuplas correctas para adquirir menciones de esas tuplas, y así entrenar un sistema tradicional de extracción de información supervisado. En este artículo analizamos las fuentes de ruido en las menciones, y exploramos métodos sencillos para filtrar menciones ruidosas. Los resultados demuestran que combinando el filtrado de tuplas por frecuencia, la información mutua y la eliminación de menciones lejos de los centroides de sus respectivas etiquetas mejora los resultados de dos modelos de extracción de información significativamente.
Palabras clave: Extracción de Información, Extracción de Relaciones, Supervisión a Distancia, Aprendizaje con Ruido

Abstract: Relation Extraction methods based on Distant Supervision rely on true tuples to retrieve noisy mentions, which are then used to train traditional supervised relation extraction methods. In this paper we analyze the sources of noise in the mentions, and explore simple methods to filter out noisy mentions. The results show that a combination of mention frequency cut-off, Pointwise Mutual Information and removal of mentions which are far from the feature centroids of relation labels is able to significantly improve the results of two relation extraction models.
Keywords: Information Extraction, Relation Extraction, Distant Supervision, Learning with Noise

1 Introduction

Distant Supervision (DS) is a semi-supervised alternative to traditional Relation Extraction (RE) that combines some of the advantages of different RE approaches. The intuition is that any sentence that contains a pair of entities that are recorded in a Knowledge Base (KB) such as DBpedia[1] or Freebase[2] to participate in a known relation (e.g., *born-in* or *film-director-of*) is likely to provide evidence for that relation. Using this approach, large training datasets of relation mentions can be automatically created by aligning entities that participate in known relations with sentences from large corpora where the entity pairs are mentioned. Such sentences are preprocessed to identify all named or numeric entities that are mentioned. Entities are identified using named entity recognizers, tagging them as persons, organizations, locations, dates, etc. If the KB specifies that a pair of entities appearing in the same sentence participates in a known relation, the corresponding textual context becomes a mention for the corresponding relation label. If the KB has no record of the two entities, the corresponding relation is marked as *unrelated* (i.e., a negative mention). Using this approach, a very large number of relation mentions can be gathered automatically, thus alleviating the sparse data problem plaguing supervised relation extraction systems, which ultimately causes overfitting and domain dependence.

In order to illustrate the method, let's consider some relations[3] and tuples from Free-

[1] http://dbpedia.org/About
[2] http://www.freebase.com/

[3] In order to improve readability, we will use intuitive tags instead of the actual Freebase relation names, i.e., *education* for */education/education/student*, *capital* for */lo-*

ISSN 1135-5948

base:

- <Albert Einstein, *education*, University of Zurich>
- <Austria, *capital*, Vienna>
- <Steven Spielberg, *director-of*, Saving Private Ryan>

Searching for the entity pairs in those tuples, we can retrieve sentences that express those relations:

- **Albert Einstein** was awarded a PhD by the **University of Zurich**.
- **Vienna**, the capital of **Austria**.
- Allison co-produced the Academy Award-winning **Saving Private Ryan**, directed by **Steven Spielberg**...

Although we show three sentences that do express the relations in the knowledge-base, distant supervision generates many noisy mentions that hurt the performance of the relation extraction system. We identified three different types of noise in the mentions gathered by distant supervision:

1. Sentences containing related entities, but which are tagged as 'unrelated' by DS. This happens because the KB we use, as all real-world Kbs, is incomplete, i.e., it does not contain all entities that participate in a given relation type.
2. Sentences containing unrelated entities, tagged as related. This happens when both participating entities that are marked as related in the KB appear in the same sentence, but the sentence does not support the relation.
3. Sentences containing a pair of related entities, but which are tagged as a mention of another, incorrect, relation. This type is the most common, and happens for entity tuples that have more than one relational label. These were previously called multi-label relations in the literature (Hoffmann et al., 2011).

Suppose that we have an incomplete KB according to whom the tuple <Brazil, Celso Amorim> is unrelated. In reality Celso is a minister of Brazil, and thus a mention of the *country-minister* relation. Mentions like *Celso Amorim , the Brazilian foreign minister , said the (...)* will be tagged by DS systems as unrelated at the training dataset, instead of appearing as *country-minister* as it should be. This is an example of Type 1.

Situations of Type 2 noise occur with tuples like <Jerrold Nadler, *born_in*, Brooklyn>. If the system extracts the following sentences from the corpora, *(...) Representative Jerrold Nadler, a Democrat who represents parts of Manhattan and Brooklyn, (...)* and *Nadler was born in Brooklyn, New York City.*, they both will be tagged as *born_in* and used later for training, although the entity tuple in the first sentence is not a positive mention of the relation under consideration.

Below we give an example of Type 3 noise. Consider the tuple <Rupert Murdock, News Corporation>. This is a multi-label relation with labels *founder* and *top-member*. Thus sentences in the training set such as *News Corporation was founded by Rupert Murdock* and *Rupert Murdock is the CEO of News Corporation* will be both considered as mentions for both *founder* and *top-member*, even though the first sentence is not a mention for the *top-member* relation and the second is not a mention for the *founder* relation.

We selected randomly 100 mentions respectively from single-label related mentions, multi-labeled related mentions and unrelated mentions which correspond to Freebase relations as gathered by (Riedel, Yao, and McCallum, 2010). We analyzed them, and estimated that around 11% of the unrelated mentions belong to Type 1, 28% of related single-labeled mentions belong to Type 2. Regarding multi-labeled mentions, 15% belong to Type 3 and 60% to Type 2, so only 25% are correct mentions. All in all, the dataset contains 91373 unrelated mentions, 2330 single-labeled and 26587 multi-labeled mentions, yielding an estimate of 29% correct instances for related mentions, and 74% correct instances overall.

Noisy mentions decrease the performance of distant supervision systems. However, because the underlying datasets are generally very large, detecting and removing noisy mentions manually becomes untenable. This paper explores several methods that automatically detect and remove noisy mentions generated through DS.

cation/country/capital, and *director-of* for */film/director/film*

2 Related Work

Distant Supervision was originally proposed by (Craven and Kumlien, 1999) for the biomedical domain, extracting binary relations between proteins, cells, diseases and more. Some years later, the approach was improved by (Mintz et al., 2009), making it available for different domains, such as *people, locations, organizations,...*, gaining popularity since then.

We can find many approaches that model the noise to help the classifier train on the respective datasets. (Riedel, Yao, and McCallum, 2010) model distant supervision for relation extraction as a multi-instance single-label problem, allowing multiple mentions for the same tuple, but it does not allow more than one label per object. (Hoffmann et al., 2011) and (Surdeanu et al., 2012) focus on multi-instance multi-label learning.

Distant supervision has also been the most relevant approach used to develop different relation extraction system at the *TAC-KBP Slot Filling* task[4] for the last years, organized by NIST. Nearly all the participants use distant supervision for their systems to extract relations for *people* and *organization* entities. The approach has improved slowly during the latest years, and working with noisy mentions to train the systems has been recognized as the most important hurdle for further improvements.

3 Distant Supervision for Relation Extraction

The methods proposed here for cleaning the textual evidence used to train a RE model are system independent. That is, they apply to any RE approach that follows the "traditional" distant supervision heuristic of aligning database tuples with text for training. As proof of concept, in this paper we use two variants of the *Mintz++* system proposed by (Surdeanu et al., 2012) and freely available at `http://nlp.stanford.edu/software/mimlre.shtml`. This algorithm is an extension of the original work of (Mintz et al., 2009) along the following lines:

- The *Mintz++* approach models each relation mention independently, whereas Mintz et al. collapsed all the mentions of the same entity tuple into a single datum. In other words, *Mintz++* constructs a *separate* classification data point from every sentence that contains a training tuple, whereas the original Mintz et al. algorithm merges the features extracted from all sentences that contain the same tuple into a single classification mention. The former approach was reported to perform better in practice by (Surdeanu et al., 2012).
- *Mintz++* allows multiple labels to be predicted for the same tuple by performing a union of all the labels proposed for individual mentions of the same tuple, whereas the Mintz et al. algorithm selected only the top-scoring label for a given entity pair. The multiple-label strategy was also adopted by other models ((Hoffmann et al., 2011); (Surdeanu et al., 2012)). This is necessary because the same pair of entities may express multiple relations, e.g., (*Balzac*, *France*) expresses at least two relations: *BornIn* and *Died*, which cannot be modeled by Mintz et al.'s algoritm.
- *Mintz++* implements a bagging strategy that combine five individual models. Each model is trained using four out of five folds of the training corpus. The final score is an unweighted average of the five individual scores. In this paper, we report results using two variants of the *Mintz++* model: when this ensemble modeling strategy is enabled (*Mintz++*) or disabled, i.e., using a single model trained over the entire training data (which we will call *Mintz**). This allows us to directly compare the effects of bagging with the impact of the data-cleanup proposed in this paper.

The results reported here are generated over the corpus created by (Riedel, Yao, and McCallum, 2010) and used by many other IE researchers like (Hoffmann et al., 2011), (Surdeanu et al., 2012), inter alia. This corpus uses Freebase as the source for distant supervision and the New York Times (NYT) corpus by (Sandhaus, 2008) for the source of textual evidence. The corpus contains two partitions: a training partition, containing 4700 relation mentions from the 2005–2006 portion of the NYT corpus, and a testing

[4]Task definition for 2013 available at `http://surdeanu.info/kbp2013/KBP2013_TaskDefinition_EnglishSlotFilling_1.0.pdf`

partition, containing 1950 more recent (2007) relation mentions. Because this corpus does not have a reserved development partition, we tuned our models over the training partition using cross-validation. In both partitions, negative mentions were automatically generated from pairs of entities that co-occur in the same sentence and are not recorded in Freebase with any relation label. Crucially, the corpus authors released only a random subsample of 5% of these negative mentions for the training partition. This means that any results measured over the training partition will be artificially inflated because the systems have fewer chances of inferring false positive labels.

4 Methods to Remove Noise

We tried three different heuristics to clean noisy mentions from the dataset. We experimented removing tuples depending on their mention frequency (MF), their pointwise mutual information (PMI), and the similarity between the centroids of all relation mentions and each individual mention (MC). We also built several ensemble strategies that combine the most successful individual strategies, as parametrized over development data. Note that none of these methods uses any additional manual annotation at all.

4.1 Mention Frequency (MF)

For our first heuristic, we consider that tuples with too many mentions are the most probable to contain noisy mentions, so we remove all those tuples that have more than a predefined number of mentions. Our system removes both positive tuples that appear in Freebase, and negative (unrelated) tuples which contain more than X mentions. We experimented with different thresholds and chose the limit that gave the highest F-Measure on the development set[5]. The chosen value was $X = 90$, i.e., all tuples with more than 90 mentions where removed, around 40% of the positive mentions, and 15% of the total dataset considering both positive and negative mentions.

For example, the tuple <European Union, *has-location*[6], Brussels> appears with 95 mentions. This tuple contains good mentions like *The European Union is headquartered in Brussels.* but also many noisy mentions like *The European Union foreign policy chief, Javier Solana, said Monday in Brussels that (...)* or *At an emergency European Union meeting of interior and justice ministers in Brussels on Wednesday, (...)* which do not explicitly say that Brussels is in the European Union, and can thus mislead the supervised RE system. This heuristic removes all instances of this tuple from the training data.

4.2 Tuple PMI

The second heuristic calculates the PMI between each entity tuple and the a relation label. Once we calculate the PMI for each tuple, we consider that the tuples which have a PMI below a defined threshold have noisy mentions, and remove them. Empirically, we observed that our system performs better if we remove only positive mentions with low PMI and keep the negative ones, regardless of their PMI value. Our system performed better with a threshold of 2.3, removing around 8% of the positive training tuples. This heuristic is inspired by the work of (Min et al., 2012).

As an example, this approach removed the tuple: <Natasha Richardson, *place-of-death*[7], Manhattan>. This tuple has only one mention: *(...) Natasha Richardson will read from 'Anna Christie,' (...) at a dinner at the Yale Club in Manhattan on Monday night.*. This mention does not support the place-of-death relation. That is, even though Natasha Richardson died in Manhattan, the mention is unrelated to that fact.

4.3 Mention Centroids (MC)

This heuristic calculates the centroid of all mentions with the same relation label, and keeps the most similar mentions to the centroids. We hypothesize that the noisy mentions are the furthest ones from their label centroids. For this experiment, we consider each mention as a vector and the features as space dimensions. We use the same features used by the DS system to build the vectors, with the frequency as the value of the feature. The centroid is built from the vectors as described in equation 1 below.

$$\vec{C_i} = \left(\frac{feat_1}{mentions_i}, \frac{feat_2}{mentions_i}, \ldots, \frac{feat_N}{mentions_i} \right) \quad (1)$$

[5] Throughout the paper, development experiments stand for cross-validation experiments on Riedel's training partition.

[6] /location/location/contains

[7] /people/deceased_person/place_of_death

where mentions_i = number of mentions for label i ($1 \leq i \leq M$), feat_j = number of appearances of feature j ($1 \leq j \leq N$) and C_i = Centroid for label i.

The similarity between a centroid and any given mention is calculated using the cosine:

$$cosine(C, M) = \frac{\vec{C} \cdot \vec{M}}{\sqrt{\vec{C} \cdot \vec{C}} \cdot \sqrt{\vec{M} \cdot \vec{M}}} \quad (2)$$

where C = Centroid and M = Mention.

We select a percentage of the most similar mentions to each centroid, and discard the rest. Our system returned the best results on development when we kept 90% of the most similar mentions of each relational label.

We do not use this heuristic for negative mentions. Empirically, we observed that this heuristic performs better if we kept all negative mentions rather than deleting any of them. This could be an artifact of the fact that only 5% of the negative mentions are included in Riedel's training dataset. Thus, sub-sampling negative mentions further yields datasets with too few negative mentions to train a discriminative model. This methods removes around 8% of the positive mentions.

As an example of the method, if we take the centroid for relation *company-founders*, the mention appearing in the sentence *(...) its majority shareholder is Steve Case, the founder of AOL* of the tuple <Steve Case, company-founders, AOL> is the most similar to the centroid of the same label. On the contrary, the mention *Ms. Tsien and Mr. Williams were chosen after a competition that began with 24 teams of architects and was narrowed to two finalists, Thom Mayne's Morphosis being the other* of the tuple <Thom Mayne, Morphosis> was correctly excluded, as the mention does not explicitly say that Thom Mayne is the founder of Morphosis.

4.4 Ensemble Models

We experimented with several ensemble models that combine the above individual strategies, in different order. The best results on development, as shown in Section 5.1, were different for *Mintz** and *Mintz++*. For the first, we first filtered using PMI, then run the MF filter, and finally applied the centroid-based filter. For the second, the best combination was to run PMI and then MF. The MC method did not provide any additional gain.

	Rec.	Prec.	F1
Mintz*	34.98	39.44	37.07
MF 90	33.19	44.49	**38.01**
PMI 2.3	34.49	40.64	37.31
MC 90%	34.81	40.31	37.33
PMI+MF+MC	32.72	46.36	**38.53**

Table 1: Development experiments using Mintz*, showing the results of each filtering method and the best combination.

	Rec.	Prec.	F1
Mintz++	34.85	41.45	37.86
MF 180	33.65	44.48	**38.45**
PMI 2.4	34.00	42.97	37.95
PMI+MF	33.25	45.57	**38.45**

Table 2: Development experiments using Mintz++, showing the results of each filtering method and the best combination.

5 Experiments

We evaluated the methods introduced in the previous section with the dataset developed by (Riedel, Yao, and McCallum, 2010). This dataset was created by aligning Freebase relations with the New York Times (NYT) corpus. They used the Stanford named entity recognizer to find entity mentions in text and constructed relation mentions only between entity mentions in the same sentences. We used the same features as (Riedel, Yao, and McCallum, 2010) for the mention classifier.

The development set was created using a three-fold cross-validation technique, similarly to (Surdeanu et al., 2012). For the formal evaluation on the test set, we only used the best ensemble models, instead of applying each method individually.

5.1 Results on the Development Corpus

The initial experiments where done using the *Mintz++* system in (Surdeanu et al., 2012) without any ensemble at the classifier. From now on, the *Mintz++* without the ensemble will be denoted as *Mintz** in this paper. Table 1 shows the results we obtained with each method. If we execute our methods individually, we get the best results with the *Mention frequency* experiment (Section 4.1), where

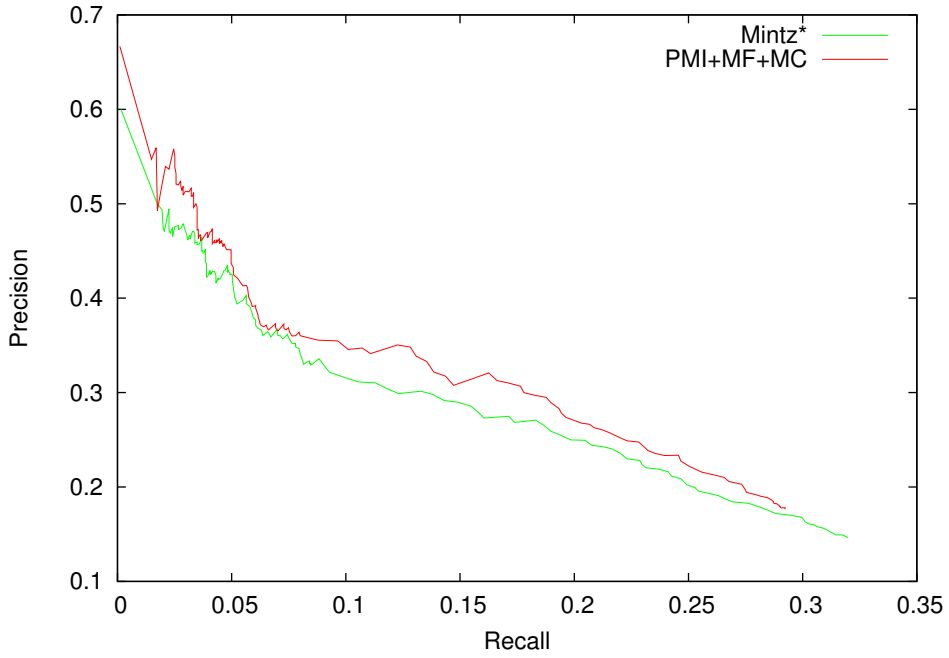

Figure 1: Precision/recall curves for the Mintz* system on the test partition. The red line is our best filtering model.

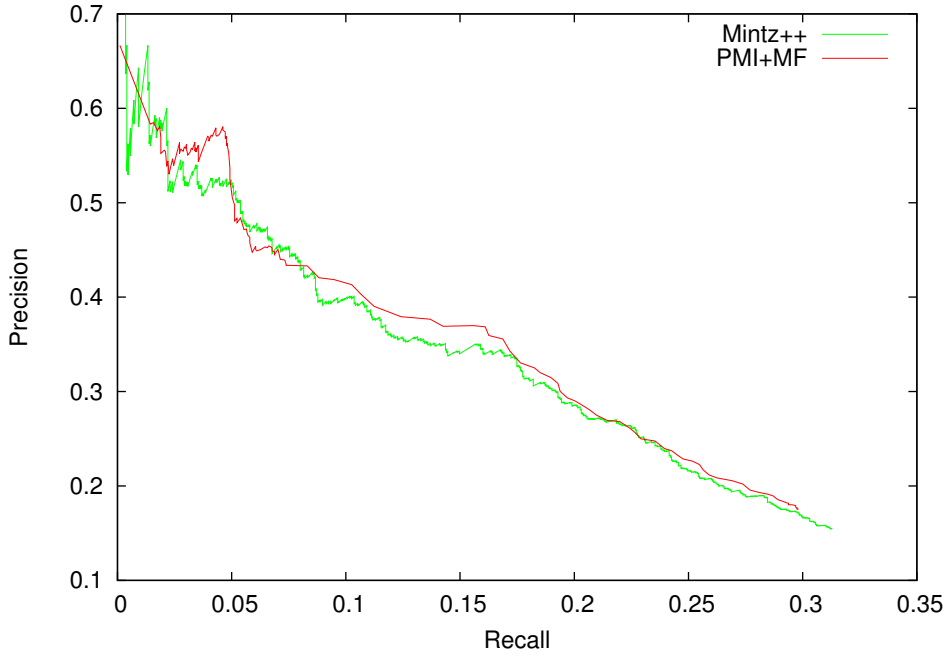

Figure 2: Precision/recall curves for the Mintz++ system on the test partition. The red line is our best filtering model.

our system's F-Measure improves nearly 1%. The PMI (Section 4.2) and the *Mention centroids* models (Section 4.3) both yield a small improvement over the baseline. For the ensemble models, we obtain the best performance by combining PMI with *Mention frequency* and the *Mention centroids*, improving the F-Measure nearly 1.5 absolute points. Our system improves the precision in each experiment, but not the recall, this scoring

	Rec.	Prec.	F1
Mintz*	31.95	14.57	20.02
PMI+MF+MC	29.23	17.64	**22.00**

Table 3: Results on the test partition for Mintz* (without bagging).

	Rec.	Prec.	F1
Mintz++	31.28	15.43	20.67
PMI+MF	29.79	17.48	**22.03**

Table 4: Results for Mintz++ (with bagging).

parameter generally decreases slightly. This is to be expected, since the models built using filtered data train on fewer positive mentions, thus they will be more conservative in predicting relation labels.

We applied the same heuristics to the original *Mintz++* system at (Surdeanu et al., 2012), and optimized them. The optimal parameters are 180 mention maximum for *Mention frequency* (4.1), and 2.4 for the *PMI* heuristics (Section 4.2). Unfortunately the *Mention centroids* (Section 4.3) heuristic did not yield an improvement here. Finally, we combined the *PMI* heuristic with the *Mention frequency* experiment to improve our results. Table 2 shows the results we obtained for each heuristic. Surprisingly, *MF 180* and *PMI+MF* give the same F-Measure.

5.2 Results on the Test Partition

For the formal evaluation on the test set, we only chose the ensemble models that performed best with the development set for *Mintz**, with the same optimal parameters obtained on development. On the test set, the F-Measure improves approximately 2 points. The results are shown in Table 3.

Figure 1 shows the precision/recall curves for our best system relative to the *Mintz** baseline. The figure shows that our approach clearly performs better.

Table 4 shows the results on the test partition of the original *Mintz++* system of (Surdeanu et al., 2012) and the *Mintz++* extended with our best ensemble filtering model (tuned on development).

Figure 2 shows the precision/recall curves of the two systems based on *Mintz++*. The models trained using filtered data perform generally better than the original system, but the differences are not as large as for the previous model that does not rely on ensemble strategies. This suggests that ensemble models, such as the bagging strategy implemented in *Mintz++*, are able to recover from some of the noise introduced by DS. However, bagging strategies are considerably more expensive to implement than our simple algorithms, which filters the data in a single pass over the corpus.

To check for statistical significance, we used the bootstrapping method proposed by (Berg-Kirkpatrick, Burkett, and Klein, 2012) verifying if the improvement provided by mention filtering is significant[8]. This bootstrapping method concluded that, although the difference between the two models is small, it is statistically significant with p-values below 0.001, thus supporting our hypothesis that data cleanup for DS algorithms is important.

6 Conclusions

Motivated by the observation that relation extraction systems based on the distant supervision approach are exposed to data that includes a considerable amount of noise, this paper presents several simple yet robust methods to remove noisy data from automatically generated datasets. These methods do not use any manual annotation at the datasets. Our methods are based on limiting the mention frequency for each tuple, calculating the Pointwise Mutual Information between tuples and relation labels, and comparing mention vectors against the mention centroids of each relation label.

We show that these heuristics, especially when combined using simple ensemble approaches, outperform significantly two strong baselines. The improvements hold even on top of a strong baseline that uses a bagging strategy to reduce sensitivity to training data noise.

References

Berg-Kirkpatrick, Taylor, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and*

[8]The statistical significance tests used the points at the end of the P/R curves.

Computational Natural Language Learning, EMNLP-CoNLL '12, pages 995–1005, Stroudsburg, PA, USA. Association for Computational Linguistics.

Craven, Mark and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press.

Hoffmann, Raphael, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 541–550, Stroudsburg, PA, USA. Association for Computational Linguistics.

Min, Bonan, Xiang Li, Ralph Grishman, and Sun Ang. 2012. New york university 2012 system for kbp slot filling. In *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*. National Institute of Standards and Technology (NIST).

Mintz, Mike, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

Riedel, Sebastian, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '10)*.

Sandhaus, Evan. 2008. The new york times annotated corpus. In *Linguistic Data Consortium, Philadelphia*.

Surdeanu, Mihai, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 455–465, Stroudsburg, PA, USA. Association for Computational Linguistics.