Saarland University
Language Science and Technology
Master's Thesis
Prof. Dr. Dietrich Klakow
Prof. Dr. Iryna Gurevych

Relation Extraction Using Liberalism love and beloved

November 28, 2013

Ehsan Khoddammohammadi

Acknowledgements

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Declaration

I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged.

Saarbrücken, 4th December 2013

Ehsan Khoddammohammadi

Abstract

Keywords: Blah, Blah

Contents

1 Linking Text to a Knowledge Base for Relation Discovery

In this chapter I introduce an idea which relates previous work on representation learning of KBs to relation extraction from text. First I propose a method which tries to learn embeddings of entities and relations both from Freebase and a corpus prepared in a specific format. With such a rich embeddings, I will propose a model to and settings to predict links between entities in Freebase or relations among entities mentioned in the corpus. Later on, the experimental setup and a pipeline created for this purpose will be described and finally in the last section we will evaluate our model and discuss different aspects of the results.

1.1 Learning Representaion of Entities and Relations from Text and Knowledge Base

Two direction of works have been conducted previously on learning representation of words which we discussed in **??**. 1) KB-based representation learning and 2)Corpus-based methods. Here I will demonstrate a process which will enable us to jointly learn embeddings with both contextual and lexical information. Among previously discussed methods, I borrow a model proposed in [1]. As I described in **??**, this model can take set of binary predicates and their arguments and induce continuous features for both predicates and arguments in vector space. Using these embeddings and a bilinear combination of them, the model can discriminate between true facts in the dataset and negative examples. For more information of this model please see **??**.

We can see that this models in its plain vanilla form is limited to learn embeddings from a KB but has a great potential to be extended in various aspects. One of the possible aspects which matters for relation extraction task, which is main motivation of this work, is that it should be able to learn facts also from text. The reason is that, despite the vast effort of

gathering information and encode it as knowledge in KBs, they are limited in the sense of coverage. To compensate this problem we need to discover new facts from corpus.

In chapter **??** we have extensively discussed major related works and we also discussed important features that were being used in many important works and are shown to be very effective. Most of these feature were previously encoded in a way that can be used for classic classifiers. I will first enumerate these features and then we will see that how can we formalized them to make it possible to use them with a neural distributed model discussed in **??**. A thorough discussion of creating dataset will appear in 1.4 but for now, it is sufficient to know that by a corpus we mean, a pre-processed clean pile of text in English, coming from harvested web pages or from news papers which contains these annotations: 1) part of speech tags 2)dependency parsed 3) named entities are recognized and tagged by their type. Our process of extracting features will be described fully in 1.4, in the next part, Any feature I describe comes from a sentence which contains two named entities.

1.1.1 Informative Features for Relation Extraction

In this part, I briefly enumerate important features which have been used extensively in previous works. For an actul example, I show some instances from a dataset prepared by Riedel et al. and used in [3] and [4]. For full description of futures in their work please see **??**.

A relevent set of features that I harvest from text to my work is as follow:

Type of Named Entities

These types are simple set of types that usually named entity taggers tag entities with them. For example: **LOC** for locations, **PER** for persons. These types contain a minimalistic information but yet useful. The actual type of entities, features of an entity which are common with similar entities should be induced by the model.

Dependency Role of Named Entities

Dependency structures have useful information and there is a long history of using dependency patterns in literature. For example we discussed [**?**], [**?**] in **??** and we showed the importance of dependency roles in these models. I follow previous works and will use this type of features in my proposed model. As an example a NE

in a senctence can have either of these roles: direct object *dobj*, passive object *pobj*, appositional modifier *appos*, participial modifier *partmod*, nominal subject *nsubj*, noun compound modifier *nn* and prepositional modifier *prep*.

Head of Sentence in Dependency Path

The dependency path between arguments and the head of the sentence worked as surface patterns indicating a relation if it appears in different contexts. All the head words are collected and I use them as surface patterns. They are labels for relations that we want to induce or predict among entities.

Another set of features, the most important one, which are available in KBs are actual knowledge about entities and the relation between them. This types of relations naturally are encoded as predicate-arguments relations. For example this relation:

/PEOPLE/PERSON/NATIONALITY(MIR-HOSSEIN MOUSAVI, IRAN)

indicates that *Mir-Hossein Mousavi* has *Iranian* nationality.

1.2 Task Description

We have a set of facts, $\mathcal{F}$, coming from a KB which any instance of it is in form of a triple (e_i, rel_k, e_j) which arguments of this triple are left and right named entities and the predicate is a relation from a certain type. Additionally we have a corpus of text, $\mathcal{C}$, which is tagged by part of speech, named entities and parsed with a dependency parser. A positive triple is a triple which at least an element of context in $\mathcal{C}$, here a sentence, supports this triple. Likewise, a negative triple is a triple that there is no support available for it in the corpus. Given $\mathcal{F}$ and $\mathcal{C}$ we would like to perform two tasks jointly:

1. Learning embeddings of named entities, KB relations and surface patterns from information in $\mathcal{F}$ and $\mathcal{C}$.
2. Learning a model with objective of ranking all positive triples lower than all negative triples, preferably with a large margin.

In the domain of machine learning there are several models with ability of performing learning representations and classification jointly which we discussed some of them in **??**. Among those models relevant to our task, we picked the neural distributed model proposed in [1]. A detail of this model is discussed in **??**.

This model takes a set of triples $\mathcal{T}$ as an input and perform both of our desired tasks. With having this model the only problem now is to generate $\mathcal{T}$ such as it staisfies our objective. In the next section I will describe a method for generating this dataset and later in **??** I emperically show the effectivness of this method.

1.3 Linking Text to KB

To benefit from information both available in a KB and hidden in the text we need to link them so information can transfer from one to the another. This can be happened through sharing the task of learning representations of words specially NEs and surface patterns. This section is dedicated to elaborate on the approach I took to share the task of representation learning between text and KB.

The main idea is to present important features of text in such a formalization that can be mutually learn with set of fact that come from a KB, for instance Freebase. We mentioned that knowledge in Freebase is encoded as triples of a predicate and two arguments. We take the same formalism and by introducing auxillary predicates, we encode the annotations of a corpus in form of triples.

These are list of auxiliary relations which we add to $\mathcal{T}$ with their definition:

HAS_TYPE this predicate takes a NE as left argument and its type produced by NE-tagger as right argument. For example:

HAS_TYPE(MIR-HOSSEIN MOUSAVI, PER)

HAS_DEP_ROLE with this predicate we relate a NE to its dependency role:

HAS_DEP_ROLE(MIR-HOSSEIN MOUSAVI, DOBJ)

Head of Dependency path for each dependency path, we can consider head of a path as a relation or a predicate which relates two NEs of the path together. This head word ,which in our case is always verb but in general can also be a nominal phrase, will work as a surface pattern and a candidate relation. From now on we will call this head words *triggers*, following the work in [**?**]. For example for this sentence and its dependency path:

Mir-Hossein Mousavi, president of Iran, said ...
PATH#APPOS|->APPOS->PRESIDENT->PREP->OF->POBJ->|POBJ

we will have the relation below in $\mathcal{T}$:

PRESIDENT (MIR-HOSSEIN MOUSAVI, IRAN)

HAS_TRIGGER we know that we have two types of relations in $\mathcal{T}$, relations that come from corpus $\mathcal{C}$ and those from a KB $\mathcal{F}$. If there are two NEs in $\mathcal{C}$ which are related by a trigger and there exist a relation between them in $\mathcal{F}$ too then we will add a third relation, **HAS$_T$RIGGER** between the knowledge base relation and a trigger relation. For example given these two relations in $\mathcal{T}$:

PRESIDENT (MIR-HOSSEIN MOUSAVI, IRAN)
/GOVERNMENT/POSITION_HELD/PRESIDENT (MIR-HOSSEIN MOUSAVI, IRAN)

we will add the third relation below to $\mathcal{T}$:

HAS_TRIGGER (/GOVERNMENT/POSITION$_h$ELD/PRESIDENT, PRESIDENT)

The reason is to increase correlation between KB relations and text surface patterns which will enable the model to transfer information across KB and corpus.

Having $\mathcal{T}$ generated now we can run our experiments. In the next section I describe the pipeline and after that we will see the evaluation of our approach.

1.4 Experimental Setup

In this section, first we will describe the process of producing a dataset I use for running experiments and then we will see different experiments I ran to examine the effectiveness of proposed approach for relation discovery.

1.4.1 Creating Dataset

There are three phases to create a suitable dataset for our experiments. In the first phase, a corpus should be annotated by part of speech and named entity tags and parsed with dependency parser. This will lead to produce our desired corpus $\mathcal{C}$. In the second phase, we pick a KB and use it as our $\mathcal{F}$ dataset. If $\mathcal{F}$ is too big in sense of size of relations for our computing resource we can limit ourselves to a subset of KB. I followed the heuristic proposed in [2] and [3] and limit $\mathcal{F}$ only to relations which their argument have been co-occurred in a sentence in $\mathcal{C}$.

In the third phase, using $\mathcal{C}$ and $\mathcal{F}$ we produce triples that we discussed in section 1.3 and create our final dataset $\mathcal{T}$.

To make our corpus we follow the protocol described in [3] and use their dataset since it is fully compatible with our requirements. They used New-York Times corpus

2 Entity Linking Among Lexical Resources

2.1 Experimental Setup

In this part of paper we describe the methodology we followed to encode available information in two different lexical resources, WordNet and GermaNet, that makes it possible to link entities of the two different resources and learn bi-lingual embeddings of word senses in German and English. The main idea is to relate two senses from two different resources using cross-lingual sense alignments. This is an additional information which can play a role of bridge between two different tasks, learning German embeddings and English embeddings, and can help to transfer knowledge from one to the another. Using this new feature we make our WordNet-GermaNet dataset which contains three type of relations (1) WordNet relations (2) GermaNet relations (3) Cross-lingual sense alignments between WordNet and GermaNet

First two types of relations are directly extracted from WordNet and GermaNet and for the cross-lingual relations we used Interlingual Index mappings between WordNet and GermaNet.
Example of relations:

WN-sense-A	WN-rel-1	WN-sense-B
GN-sense-C	GN-rel-2	GN-sense-D
WN-sense-A	ILI-rel-1-2	WN-sense-B

Left and right entities are WordNet and GermaNet senses and relations are current semantical relations in each of lexicons such as: meronymy, holonymy and ....

We have created four different dataset, each divided to train, test and validation separated subsets. Our four datasets are:

1. Only WordNet triples (WN)
2. Only GermaNet triples (GN)
3. WordNet-GermaNet triples with one-direction cross-lingual alignments (WN-GN)
4. WordNet-GermaNet with double-direction cross-lingual alignments (WN-GN DD)

Dataset 3 includes both relations extracted from WordNet and GermaNet and also the mapping between senses. Dataset 4 is same as dataset 3 but since the models we will use are assuming all the relations are assymetric we will try to encode the symmetry of cross-lingual alignments by reversing each of them and include the reverse in the dataset. Datasets 3 and 4 contains two different variants: the first variants contains only WordNet relations (test on WN) in the held-out test dataset and the second variant contains only GermaNet triples (test on GN). In this way we can observe the direction of possibly transferring information from English to German or vice versa.

For reducing the sparsity of data and boosting the learning runtime we filtered out all the entities that appeared less than 3 times in our datasets.

(version of wordnet, germanet, ILI and role of uby should be described here)

2.2 Evaluation

To show the effectiveness of joint learning of features from multiple knowledge bases we suggest two experiment setups. In the first schema we follow Bordes et al. ranking task. The goal of this task is to show how well the information in knowledge bases can be preserved by the learned features. On the other hand, the second setup is investigating on this question that if the learned word embeddings from multiple resources are able to improve the performance of monolingual embeddings in a standard NLP task, here word-pair similarity or not. In this setup we will look to contribution of the learned features in predicting similarity of words.

2.2.1 Evaluation Using Reconstruction

Bordes et al. (Bordes2011) proposed a ranking task that for each triple (e_i, r_k, e_j) in the data set, all the entities will be ranked as a candidate for being right entity of the triple

given the relation and the left entity. Depends on which one of the models is used, SE or SME-Bil, all the entities will be sorted based on their score regarding Equation (**??**) or Equation **??** previously introduced in section **??**. By keeping the statistics of difference between the predicted rank of e_j and its true rank and also repeat the same process for left entities, we will be able to report the mean and median predicted rank of entities per relation and in total. Bordes et al. proposed to schema for calculating the average rank, micro averaging which emphasis on more frequent relations by weighted averaging with frequency of relations as weights and macro averaging which consider all the relations equally , either frequent or infrequent ones. The third statistic that we report following their work, r@100, is the ratio of number of times that an entity is correctly among top 100 entities ranked and predicted for a triple to the number of occurances of this entity in the dataset. We applied SE and SME-Bil models on our created datasets and the ranking performance on each of them is presented in Table 2.1.

2.2.1.1 Evaluation on feature informativity

We are interested to further analyze the effectiveness of learned embeddings to capture semantic features of words, therefor we compare the embeddings learned a single resource or from multiple resources against human judgments. Five datasets of word-pair similarity are used to compare the correlation of predicted similairty of pair of words against human judgments. [rubensteinGoodenough], [yangPowers], [millerCharles],[Szumlanski] and [finkelstein] are English datasets that we used to meaure the correlation of similarities predicted by our embeddings and embeddings induced by the other methods to human judgments. For German, we use [this and that]. The other embeddings which are used in our comparison are (Turian et al., HLBL and Klementiev et al.). To measure the similarity between any given wordpair (w_1, w_2) we find all vectors associated to different senses of the given words in our embedding dictionary and pick the pair of embeddings that maximize cosine similarity between two words. We can motivate this by saying that for each word pair any of words works as a context for disambiguating the sense of the other word.

Both Pearson and Spearman correlation of predicted and gold similarities are calculated and is reported in table 2.3 for English and 2.4 for German.

For English, we can see that ....

On the other hand in German ....

As we see in the table 2.3 in two datasets the performance of learned embeddings from bi-lingual resources are slightly worse but comparable to the mono-lingual embeddings and in the other two datasets one can observe a significant increase of performance of bi-lingual resources over monolingual resources.

More analysis on why some dataset is good and some is not good.

Table 2.1: Intrinsic Evaluation (Ranking Score Performance)

Dataset	#relations	#entities		Micro	Macro
GN SE	16	64025	lhs	84.42	72.59
			rhs	84.04	72.38
			mean	1003.59	3739.85
			median	5.0	2213.37
			global	84.23	72.49
GN SME-BIL	16	64025	lhs	79.06	58.58
			rhs	83.30	81.11
			mean	407.90	308.01
			median	10.0	54.18
			global	81.18	69.85
WN SE	23	148976	lhs	91.90	89.47
			rhs	92.30	90.25
			mean	148.72	623.10
			median	5.0	4.69
			global	92.10	89.86
WN SME-BIL	23	148976	lhs	83.08	72.21
			rhs	85.2	77.92
			mean	128.82	511.21
			median	10.0	26.63
			global	84.14	75.57
WN-GN SE (WN held out)	32	213002	lhs	90.82	89.14
			rhs	91.56	88.76
			mean	293.16	1356.30
			median	5.0	5.10
			global	91.19	88.95
WN-GN SME-BIL(WN held out)	32	213002	lhs	82.42	73.65
			rhs	83.40	73.44
			mean	124.85	331.82
			median	11.0	33.86
			global	82.91	73.55
WN-GN SE (GN held out)	32	213002	lhs	81.82	70.56
			rhs	79.92	70.06
			mean	3031.44	15470.56
			median	7.0	10080.5
			global	80.87	70.313
WN-GN SME-BIL(GN held out)	32	213002	lhs	63.54	41.64
			rhs	64.78	70.32
			mean	984.79	1021.37
			median	40.0	428.90
			global	64.16	55.98
WordNet-GermaNet-DD (GN held out)	32 11	213002	lhs	57.72	38.063
			rhs	60.72	63.617
			mean	932.49	719.47
			median	56.0	175.56
			global	59.22	50.84
WordNet-GermaNet-DD (WN held out)	32	213002	lhs	69.66	59.54
			rhs	66.60	58.95
			mean	166.18	18.0
			median	466.91	55.41
			global	68.13	59.25

Table 2.2: Ranking Performance for Mapped Relations

Dataset	#dimension	#relations	#entities		Micro(%)	Macro(%)
GermaNet	25	10	64025	lhs	82.60	68.18
				rhs	81.90	68.84
				global	82.25	68.51
WordNet	25	19	148976	lhs	83.50	83.17
				rhs	84.22	83.64
				global	83.86	83.40
WordNet-GermaNet (WN)	25	24	213002	lhs	78.70	82.60
				rhs	79.56	83.06
				global	79.13	82.83
WordNet-GermaNet (GN)	25	24	213002	lhs	69.66	59.54
				rhs	66.60	58.95
				global	68.13	59.25

Table 2.3: Word-pair Similarity Performance for English

Dataset		WN-SE	WN-GN-SE	WN-SME-BIL	WN-GN-SME-BIL	WN-GN-SME-BIL-DD	H
RubensteinGoodenough65	P	0.682	0.666	0.540	0.508	0.611	-0
	S	0.769	0.741	0.447	0.478	0.552	-
MillerCharles30	P	0.611	0.644	0.592	0.555	0.541	-0
	S	0.720	0.648	0.564	0.561	0.468	-
Finkelstein353	P	0.179	0.206	0.272	0.208	0.193	0
	S	0.087	0.146	0.240	0.196	0.162	0
Szumlanski122	P	-0.145	0.032	0.010	0.043	0.048	-0
	S	-0.159	0.034	0.035	0.037	0.041	-0
YangPowers130	P	0.729	0.682	0.597	0.767	0.819	0
	S	0.829	0.853	0.483	0.793	0.836	0

Table 2.4: Word-pair Similarity Performance for German

Dataset		GN-SE	WN-GN-SE	GN-SME-BIL	WN-GN-SME-BIL	WN-GN-SME-BIL-DD	Klementiev*
wortpaare222	P	-0.022	0.112	0.058	0.103	0.203	0.118
	S	-0.100	0.225	0.230	0.091	0.195	0.153
wortpaare30	P	0.865	0.984	-0.443	0.671	0.656	-0.887
	S	1.0	1.0	-0.500	0.682	0.686	-1.0
wortpaare350	P	-0.089	0.045	0.163	0.300	0.256	0.168
	S	-0.158	-0.017	0.135	0.295	0.231	0.117
wortpaare65	P	0.800	0.558	-0.572	0.607	0.480	0.233
	S	0.800	0.800	-0.8	0.588	0.439	0.200

Bibliography

[1] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Artificial Intelligence and Statistics (AISTATS)*, volume 22, 2012.

[2] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *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, pages 1003 – 1011, 2009.

[3] Sebastian Riedel and Limin Yao. Relation Extraction with Matrix Factorization and Universal Schemas. *Proceedings of NAACL- ...*, (June):74–84, 2013.

[4] Sebastian Riedel, Limin Yao, and A McCallum. Modeling relations and their mentions without labeled text. *Machine Learning and Knowledge ...*, pages 148–163, 2010.