Expert Systems With Applications 114 (2018) 34–45

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Joint entity recognition and relation extraction as a multi-head selection problem

Giannis Bekoulis*, Johannes Deleu, Thomas Demeester, Chris Develder

Department of Information Technology, Ghent University – IMEC, IDLab, Technologiepark Zwijnaarde 15, Ghent 9052, Belgium

ARTICLE INFO

Article history:
Received 20 April 2018
Revised 13 July 2018
Accepted 14 July 2018
Available online 17 July 2018

Keywords:
Entity recognition
Relation extraction
Multi-head selection
Joint model
Sequence labeling

ABSTRACT

State-of-the-art models for joint entity recognition and relation extraction strongly rely on external natural language processing (NLP) tools such as POS (part-of-speech) taggers and dependency parsers. Thus, the performance of such joint models depends on the quality of the features obtained from these NLP tools. However, these features are not always accurate for various languages and contexts. In this paper, we propose a joint neural model which performs entity recognition and relation extraction simultaneously, without the need of any manually extracted features or the use of any external tool. Specifically, we model the entity recognition task using a CRF (Conditional Random Fields) layer and the relation extraction task as a multi-head selection problem (i.e., potentially identify multiple relations for each entity). We present an extensive experimental setup, to demonstrate the effectiveness of our method using datasets from various contexts (i.e., news, biomedical, real estate) and languages (i.e., English, Dutch). Our model outperforms the previous neural models that use automatically extracted features, while it performs within a reasonable margin of feature-based neural models, or even beats them.

1. Introduction

The goal of the entity recognition and relation extraction is to discover relational structures of entity mentions from unstructured texts. It is a central problem in information extraction since it is critical for tasks such as knowledge base population and question answering.

The problem is traditionally approached as two separate subtasks, namely (i) named entity recognition (NER) (Nadeau & Sekine, 2007) and (ii) relation extraction (RE) (Bach & Badaskar, 2007), in a pipeline setting. The main limitations of the pipeline models are: (i) error propagation between the components (i.e., NER and RE) and (ii) possible useful information from the one task is not exploited by the other (e.g., identifying a *Works for* relation might be helpful for the NER module in detecting the *type* of the two entities, i.e., *PER*, *ORG* and vice versa). On the other hand, more recent studies propose to use joint models to detect entities and their relations overcoming the aforementioned issues and achieving state-of-the-art performance (Li & Ji, 2014; Miwa & Sasaki, 2014).

The previous joint models heavily rely on hand-crafted features. Recent advances in neural networks alleviate the issue of manual feature engineering, but some of them still depend on NLP tools (e.g., POS taggers, dependency parsers). Miwa and Bansal (2016) propose a Recurrent Neural Network (RNN)-based joint model that uses a bidirectional sequential LSTM (Long Short Term Memory) to model the entities and a tree-LSTM that takes into account dependency tree information to model the relations between the entities. The dependency information is extracted using an external dependency parser. Similarly, in the work of Li, Zhang, Fu, and Ji (2017) for entity and relation extraction from biomedical text, a model which also uses tree-LSTMs is applied to extract dependency information. Gupta, Schütze, and Andrassy (2016) propose a method that relies on RNNs but uses a lot of hand-crafted features and additional NLP tools to extract features such as POS-tags, etc. Adel and Schütze (2017) replicate the context around the entities with Convolutional Neural Networks (CNNs). Note that the aforementioned works examine pairs of entities for relation extraction, rather than modeling the whole sentence directly. This means that relations of other pairs of entities in the same sentence — which could be helpful in deciding on the relation *type* for a particular pair — are not taken into account. Katiyar and Cardie (2017) propose a neural joint model based on LSTMs where they model the whole sentence at once, but still they do not have a principled way to deal with multiple relations. Bekoulis, Deleu, Demeester, and Develder (2018) introduce

* Corresponding author.
E-mail addresses: giannis.bekoulis@ugent.be (G. Bekoulis), johannes.deleu@ugent.be (J. Deleu), thomas.demeester@ugent.be (T. Demeester), chris.develder@ugent.be (C. Develder).

https://doi.org/10.1016/j.eswa.2018.07.032
0957-4174/

a quadratic scoring layer to model the two tasks simultaneously. The limitation of this approach is that only a single relation can be assigned to a token, while the time complexity for the entity recognition task is increased compared to the standard approaches with linear complexity.

In this work, we focus on a new general purpose joint model that performs the two tasks of entity recognition and relation extraction simultaneously, and that can handle multiple relations together. Our model achieves state-of-the-art performance in a number of different contexts (i.e., news, biomedical, real estate) and languages (i.e., English, Dutch) without relying on any manually engineered features nor additional NLP tools. In summary, our proposed model (which will be detailed next in Section 3) solves several shortcomings that we identified in related works (Section 2) for joint entity recognition and relation extraction: (i) our model does not rely on external NLP tools nor hand-crafted features, (ii) entities and relations within the same text fragment (typically a sentence) are extracted simultaneously, where (iii) an entity can be involved in multiple relations at once.

Specifically, the model of Miwa and Bansal (2016) depends on dependency parsers, which perform particularly well on specific languages (i.e., English) and contexts (i.e., news). Yet, our ambition is to develop a model that generalizes well in various setups, therefore using only automatically extracted features that are learned during training. For instance, Miwa and Bansal (2016) and Li et al. (2017) use exactly the same model in different contexts, i.e., news (ACE04) and biomedical data (ADE), respectively. Comparing our results to the ADE dataset, we obtain a 1.8% improvement on the NER task and $\sim$3% on the RE task. On the other hand, our model performs within a reasonable margin ($\sim$0.6% in the NER task and $\sim$1% on the RE task) on the ACE04 dataset without the use of pre-calculated features. This shows that the model of Miwa and Bansal (2016) strongly relies on the features extracted by the dependency parsers and cannot generalize well into different contexts where dependency parser features are weak. Comparing to Adel and Schütze (2017), we train our model by modeling all the entities and the relations of the sentence at once. This type of inference is beneficial in obtaining information about neighboring entities and relations instead of just examining a pair of entities each time. Finally, we solve the underlying problem of the models proposed by Katiyar and Cardie (2017) and Bekoulis, Deleu, Demeester, and Develder (2017), who essentially assume classes (i.e., relations) to be mutually exclusive: we solve this by phrasing the relation extraction component as a multi-label prediction problem.[1]

To demonstrate the effectiveness of the proposed method, we conduct the largest experimental evaluation to date (to the best of our knowledge) in jointly performing both entity recognition and relation extraction (see Sections 4 and 5), using different datasets from various domains (i.e., news, biomedical, real estate) and languages (i.e., English, Dutch). Specifically, we apply our method to four datasets, namely ACE04 (news), Adverse Drug Events (ADE), Dutch Real Estate Classifieds (DREC) and CoNLL'04 (news). Our method outperforms all state-of-the-art methods that do not rely on any additional features or tools, while performance is very close (or even better in the biomedical dataset) compared to methods that do exploit hand-engineered features or NLP tools.

[1] *Note* that another difference is that we use a CRF layer for the NER part, while Katiyar and Cardie (2017) uses a softmax and Bekoulis et al. (2017) uses a quadratic scoring layer; see further, when we discuss performance comparison results in Section 5.

2. Related work

The tasks of entity recognition and relation extraction can be applied either one by one in a pipeline setting (Bekoulis et al., 2017; Fundel, Küffner, & Zimmer, 2007; Gurulingappa, Mateen-Rajput & Toldo, 2012) or in a joint model (Bekoulis et al., 2018; Miwa & Bansal, 2016; Miwa & Sasaki, 2014). In this section, we present related work for each task (i.e., named entity recognition and relation extraction) as well as prior work into joint entity and relation extraction.

2.1. Named entity recognition

In our work, NER is the first task which we solve in order to address the end-to-end relation extraction problem. A number of different methods for the NER task that are based on hand-crafted features have been proposed, such as CRFs (Lafferty, McCallum, & Pereira, 2001), Maximum Margin Markov Networks (Taskar, Guestrin, & Koller, 2003) and support vector machines (SVMs) for structured output (Tsochantaridis, Hofmann, Joachims, & Altun, 2004), to name just a few. Recently, deep learning methods such as CNN- and RNN-based models have been combined with CRF loss functions (Collobert et al., 2011; Huang, Xu, & Yu, 2015; Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016; Ma & Hovy, 2016) for NER. These methods achieve state-of-the-art performance on publicly available NER datasets without relying on hand-crafted features.

2.2. Relation extraction

We consider relation extraction as the second task of our joint model. The main approaches for relation extraction rely either on hand-crafted features (Kambhatla, 2004; Zelenko, Aone, & Richardella, 2003) or neural networks (Socher, Huval, Manning, & Ng, 2012; Zeng, Liu, Lai, Zhou, & Zhao, 2014). Feature-based methods focus on obtaining effective hand-crafted features, for instance defining kernel functions (Culotta & Sorensen, 2004; Zelenko et al., 2003) and designing lexical, syntactic, semantic features, etc. (Kambhatla, 2004; Rink & Harabagiu, 2010). Neural network models have been proposed to overcome the issue of manually designing hand-crafted features leading to improved performance. CNN- (dos Santos, Xiang, & Zhou, 2015; Xu, Feng, Huang, & Zhao, 2015; Zeng et al., 2014) and RNN-based (Socher, Chen, Manning, & Ng, 2013; Xu, Mou et al., 2015; Zhang & Wang, 2015) models have been introduced to automatically extract lexical and sentence level features leading to a deeper language understanding. Vu, Adel, Gupta, and Schütze (2016) combine CNNs and RNNs using an ensemble scheme to achieve state-of-the-art results.

2.3. Joint entity and relation extraction

Entity and relation extraction includes the task of (i) identifying the entities (described in Section 2.1) and (ii) extracting the relations among them (described in Section 2.2). Feature-based joint models (Kate & Mooney, 2010; Li & Ji, 2014; Miwa & Sasaki, 2014; Yang & Cardie, 2013) have been proposed to simultaneously solve the entity recognition and relation extraction (RE) subtasks. These methods rely on the availability of NLP tools (e.g., POS taggers) or manually designed features and thus (i) require additional effort for the data preprocessing, (ii) perform poorly in different application and language settings where the NLP tools are not reliable, and (iii) increase the computational complexity. In this paper, we introduce a joint neural network model to overcome the aforementioned issues and to automatically perform end-to-end relation extraction without the need of any manual feature engineering or the use of additional NLP components.