NEUROSENT-PDI at SemEval-2018 Task 7: Discovering Textual Relations With a Neural Network Model

Mauro Dragoni
Fondazione Bruno Kessler
Via Sommarive 18
Povo, Trento, Italy
dragoni@fbk.eu

Abstract

Discovering semantic relations within textual documents is a timely topic worthy of investigation. Natural language processing strategies are generally used for linking chunks of text in order to extract information that can be exploited by semantic search engines for performing complex queries. The scientific domain is an interesting area where these techniques can be applied. In this paper, we describe a system based on neural networks applied to the SemEval 2018 Task 7. The system relies on the use of word embeddings for composing the vectorial representation of text chunks. Such representations are used for feeding a neural network aims to learn the structure of paths connecting chunks associated with a specific relation. Preliminary results demonstrated the suitability of the proposed approach encouraging the investigation of this research direction.

1 Introduction

One of the emerging trends of natural language technologies is their application to scientific literature. There is a constant increase in the production of scientific papers and experts are faced with an explosion of information that makes it difficult to have an overview of the state of the art in a given domain. Recent works from the semantic web, scientometry, and natural language processing (NLP) communities aimed to improve the access to scientific literature, in particular to answer queries that are currently beyond the capabilities of standard search engines. Examples of such queries include finding all papers that address a given problem in a specific way, or to discover the roots of a certain idea.

The NLP tasks that underlie intelligent processing of scientific documents are those of information extraction: identifying concepts and recognizing the semantic relation that holds between them. Information extraction from corpora including relation extraction and classification normally involves a complicated multiple-step process.

In this paper, we present a neural network strategy for addressing the challenge of extracting informative entities from scientific papers and inferring their semantic relation among a set of six alternatives. This challenge is part of the SemEval 2018 Task 7 (Gábor et al., 2018). One of the pillar of the proposed approach is that the word embeddings used for representing the text within the neural network have been generated from a corpus containing only scientific papers instead of a general purpose one, like news repositories or Wikipedia.

2 System Implementation

NeuroSent has been entirely developed in Java with the support of the Deeplearning4j library [1] and it is composed by following two main phases:

- Generation of Word vectors (Section 2.1): raw text, appropriately tokenized using the Stanford CoreNLP Toolkit, is provided as input to a 2-layers neural network implementing the skip-gram approach with the aim of generating word vectors.

- Learning of Relations Model (Section 2.2): word vectors are used for training a recurrent neural network (RNN) (Gelenbe, 1993) with an output layer containing one node for each type of relation supported by the model. We decided to use RNN due to the necessity of working with input information provided through a sequence of input instances (i.e. the ordered arrays of embeddings corresponding to each word of the text to analyze).

[1] https://deeplearning4j.org/

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 848–852
New Orleans, Louisiana, June 5–6, 2018.

In the following subsections, we describe in more detail each phase by providing also the settings used for managing our data.

2.1 Generation of Word Vectors

The generation of the word vectors has been performed by applying the skip-gram algorithm on the raw natural language text extracted from a collection of 2,459,264 scientific papers collected from proceedings of past conferences and journals. In particular, we used the text extracted from the papers contained within the ACM Digital library, IEEE Xplore and Springer LNCS website. The rationale behind the choice of this dataset focuses on two reasons:

- the dataset contains only scientific documents. This way, we are able to build word embeddings focused on the scientific context.
- the dataset is smaller with respect to other corpora used in the literature for building other word embeddings that are currently freely available, like the Google News ones. [2] Indeed, as introduced in Section 1, one of our goal is to demonstrate how we can leverage the use of dedicated resources for generating word embeddings, instead of corpora's size, for improving the effectiveness of classification systems.

These two points represent the main original contributions of this work, in particular the aspect of considering only scientific information for generating word embeddings. While embeddings currently available are created from big corpora of general purpose texts (like news archives or Wikipedia pages), ours are generated by using a smaller corpus containing documents strongly related to the problem that the model will be thought for. On the one hand, this aspect may be considered a limitation of the proposed solution due to the requirement of training a new model in case of problem change. However, on the other hand, the usage of dedicated resources would lead to the construction of more effective models.

Word embeddings have been generated by the Word2Vec implementation integrated into the Deeplearning4j library. The algorithm has been set up with the following parameters: the size of the vector to 64, the size of the window used as input of the skip-gram algorithm to 5, and the minimum word frequency was set to 1. The reason for which we kept the minimum word frequency set to 1 is to avoid the loss of rare but important words that can occur in specific documents.

[2] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

2.2 Learning of The Relations Model

The relations model is built by starting from the word embeddings generated during the previous phase.

The first step consists in converting each textual sentence contained within the dataset into the corresponding numerical matrix $\mathbf{S}$. Given a sentence s, we extract all tokens t_i, with $i \in [0, n]$, and we replace each t_i with the corresponding embedding $\mathbf{w}$. During the conversion of each word in its corresponding embedding, if such embedding is not found, the word is discarded. At the end of this step, each sentence contained in the training set is converted in a matrix $\mathbf{S} = [\mathbf{w}^{\langle 1 \rangle}, \ldots, \mathbf{w}^{\langle n \rangle}]$.

Before giving all matrices as input to the neural network, we need to include both padding and masking vectors in order to train our model correctly. Padding and masking allows us to support different training situations depending on the number of the input vectors and on the number of predictions that the network has to provide at each time step. In our scenario, we work in a many-to-one situation where our neural network has to provide one prediction as result of the analysis of many input vectors (word embeddings).

Padding vectors are required because we have to deal with the different length of sentences. Indeed, the neural network needs to know the number of time steps that the input layer has to import. This problem is solved by including, if necessary, into each matrix $\mathbf{S}_k$, with $k \in [0, z]$ and z the number of sentences contained in the training set, null word vectors that are used for filling empty word's slots. These null vectors are accompanied by a further vector telling to the neural network if data contained in a specific positions has to be considered as an informative embedding or not.

A final note concerns the back propagation of the error. Training recurrent neural networks can be quite computationally demanding in cases when each training instance is composed by many time steps. A possible optimization is the use of truncated back propagation through time (BPTT) (Werbos, 1990) that was developed for re-

ducing the computational complexity of each parameter update in a recurrent neural network. On the one hand, this strategy allows to reduce the time needed for training our model. However, on the other hand, there is the risk of not flowing backward the gradients for the full unrolled network. This prevents the full update of all network parameters. For this reason, even if we work with recurrent neural networks, we decided to do not implement a BPTT approach but to use the default backpropagation implemented into the DL4J library.

Concerning information about network structure, the input layer was composed by 64 neurons (i.e. embedding vector size), the hidden RNN layer was composed by 128 nodes, and the output layer contained 6 nodes, one for each relation (described in Section 3). The network has been trained by using the Stochastic Gradient Descent with 1000 epochs and a learning rate of 0.002.

3 The Tasks

The SemEval 2018 Task 7 is composed by two different subtasks concerning the identification and classification of semantic relations. For the first subtask the participants had to identify pairs of entities that are instances of any of the six semantic relations. This was an extraction task. While for the second subtask, participants had to classify extracted instances into one of the six semantic relation types. This was a classification task.

The six semantic relations subject of this task are the following.

- **USAGE**: this is an asymmetrical relation holds between two entities X and Y, where, for example: "X is used for Y".

- **RESULT**: this is an asymmetrical relation holds between two entities X and Y, where, for example: "X gives as a result Y" (where Y is typically a measure of evaluation).

- **MODEL-FEATURE**: this is an asymmetrical relation holds between two entities X and Y, where, for example: "X is a feature/an observed characteristic of Y".

- **PART_WHOLE**: this is an asymmetrical relation holds between two entities X and Y, where, for example: "X is a part, a component of Y".

- **TOPIC**: this is an asymmetrical relation holds between two entities X and Y, where, for example: "X deals with topic Y" or "X (author, paper) puts forward Y" (an idea, an approach).

- **COMPARE**: this is a symmetrical relation holds between two entities X and Y, where "X is compared to Y" (e.g. two systems, two feature sets or two results).

Below, we briefly described the two subtasks composing the SemEval 2018 Task 7.

Subtask #1: Relation classification The subtask is decomposed into two scenarios according to the data used: classification on clean data and classification on noisy data.

For this subtask, instances with directionality are provided in both the training data and the test data and they are not to be modified or completed in the test data.

This subtask was then split in two separated activities.

1.1 *Relation classification on clean data*. The classification task is performed on data where entities were manually annotated following the ACL RD-TEC 2.0 guidelines [3]. Entities represent domain concepts specific to NLP, while high-level scientific terms (e.g. "hypothesis", "experiment") are not annotated.

1.2 *Relation classification on noisy data*. This activity is identical to the previous one with the difference is that the entities are annotated automatically and contain noise. The annotation comes from the ACL-RelAcS corpus [4] (Gábor et al., 2016) and it is based on a combination of automatic terminology extraction and external ontologies. Entities are therefore terms specific to the given corpus, and include high-level terms (e.g. "algorithm", "paper", "method"). Relations were manually annotated in the training data and in the gold standard, between automatically annotated entities.

Subtask #2: Relation extraction and classification This subtask combines the extraction task and the classification task. The training data for

[3] https://lipn.univ-paris13.fr/ gabor/Relacs/

[4] http://pars.ie/publications/papers/pre-prints/acl-rd-tec-guidelines-ver2.pdf

this scenario is the same that is used for the previous subtask, i.e. manually annotated entities, semantic relations with relation types between these entities. The test data contains different abstracts than the previous subtask and only entity annotations were provided. For the extraction task, participants need to identify pairs of entities in the abstracts that correspond to an instance of any of the six relations. While, for the classification task, relation labels of the extracted relations need to be predicted similarly to Subtask #1.

The NeuroSent system has been applied to both subtasks. In Section 4, we report the preliminary results obtained by NeuroSent on the training set compared with a set of baselines.

4 In-Vitro Evaluation

Approach	Task #1.1	Task #1.2
Support Vector Machine	0.4534	0.4551
Naive-Bayes	0.4788	0.4787
Maximum Entropy	0.4917	0.4892
CNN Architecture	0.5329	0.4918
NeuroSent	**0.5572**	**0.5749**

Table 1: Results obtained on the training set by NeuroSent and by the four baselines for the Task#1.

Approach	Task #2.1	Task #2.2
Support Vector Machine	0.3591	0.3498
Naive-Bayes	0.3842	0.3658
Maximum Entropy	0.3982	0.3755
CNN Architecture	0.4103	0.3814
NeuroSent	**0.4274**	**0.4009**

Table 2: Results obtained on the training set by NeuroSent and by the four baselines for the Task#2.

Approach	Task #1.1	Task #1.2
ETH-DS3Lab	0.817	0.904
NeuroSent	0.180	0.218

Table 3: Results obtained on the test set by NeuroSent and by the best system of Task#1.

The NeuroSent approach have been preliminarily evaluated by adopting the Dranziera protocol (Dragoni et al., 2016). This protocol, even if it was thought for the sentiment analysis task, can be easily adapted to any NLP task.

Approach	Task #2.1	Task #2.2
ETH-DS3Lab	0.488	0.493
UWNLP	0.500	0.391
NeuroSent	0.256	0.031

Table 4: Results obtained on the test set by NeuroSent and by the best systems of Task#2.

The validation procedure leverage on a five-fold cross evaluation setting in order to validate the robustness of the proposed solution. The approach has been compared with four baselines:

- Support Vector Machine (SVM): classification was run with a linear kernel type by using the Libsvm (Chang and Lin, 2011). Libsvm uses a sparse format so that zero values do not need to be captured for training files. This can cause training time to be longer, but keeps Libsvm flexible for sparse cases.

- Naive Bayes (NB) and Maximum Entropy (ME): the MALLET: MAchine Learning for LanguagE Toolkit (McCallum, 2002) was used for classification by using both Naive Bayes and Maximum Entropy algorithms. For the experiments conducted in our evaluation, the Maximum Entropy classification has been performed by using a Gaussian prior variance of 1.0.

- Convolutional Neural Network (Chaturvedi et al., 2016) (CNN): we compared our architecture with a classic CNN. Models have been trained with the embeddings created from the Blitzer dataset.

Tables 1 and 2 show the results obtained on Tasks #1 and #2 respectively. In each table, we provide averaged F1-Score obtained on the five folds in which the training set has been split.

We performed a detailed error analysis concerning the performance of NeuroSent. In general, we observed how our strategy tends to provide false negative predictions. Unfortunately, on the test set our approach obtained significant worse results with respect to the other systems participated to the competition (Tables 3 and 4). We are investigating about the reasons of these low performance.

On the one hand, a possible action for improving the effectiveness our strategy is to increase the granularity of the embeddings (i.e. augmenting the size of the embedding vectors) in order

to increase the distance between the space regions of each kind of relation. On the other hand, by increasing the size of embedding vectors, the computational time for building, or updating, the model and for evaluating a single instance increases as well. Part of the future work, will be the analysis of more efficient neural network architectures able to manage augmented embedding vectors without negatively affecting the efficiency of the platform.

5 Conclusion

In this paper, we described the NeuroSent system presented at SemEval 2018 Task 7. Our system makes use of artificial neural networks to extract relevant text chunk from scientific documents and to label pairs of them with semantic relation tags. Obtained results demonstrated the suitability of NeuroSent with respect to the adopted baselines. We may also observed how solutions based on neural networks obtained a significant improvement with respect to the others for both tasks. Future work will focus on improving the system by exploring the integration of knowledge bases (Dragoni et al., 2015) in order to move toward a more cognitive approach.

References

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM TIST*, 2(3):27:1–27:27.

Iti Chaturvedi, Erik Cambria, and David Vilares. 2016. Lyapunov filtering of objectivity for spanish sentiment model. In *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 4474–4481. IEEE.

Mauro Dragoni, Andrea Tettamanzi, and Célia da Costa Pereira. 2016. DRANZIERA: an evaluation protocol for multi-domain opinion mining. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Mauro Dragoni, Andrea G. B. Tettamanzi, and Célia da Costa Pereira. 2015. Propagating and aggregating fuzzy polarities for concept-level sentiment analysis. *Cognitive Computation*, 7(2):186–197.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 Task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Kata Gábor, Haïfa Zargayouna, Davide Buscaldi, Isabelle Tellier, and Thierry Charnois. 2016. Semantic annotation of the ACL anthology corpus for the automatic analysis of scientific literature. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Erol Gelenbe. 1993. Learning in the recurrent random neural network. *Neural Computation*, 5(1):154–164.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. `http://mallet.cs.umass.edu`.

P. J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.