Neural Semantic Role Labeling using Verb Sense Disambiguation

Domenico Alfano
Eustema S.p.A.
d.alfano@eustema.it

Roberto Abbruzzese
Eustema S.p.A.
r.abbruzzese@eustema.it

Donato Cappetta
Eustema S.p.A.
d.cappetta@eustema.it

Abstract

The Natural Language Processing (NLP) community has recently experienced a growing interest in Semantic Role Labeling (SRL). The increased availability of annotated resources enables the development of statistical approaches specifically for SRL. This holds potential impact in NLP applications. We examine and reproduce the Marcheggiani's system and its individual components, including its annotated resources, parser, classification system, the features used and the results obtained by the system.
Then, we explore different solutions in order to achieve better results by approaching to Verb-Sense Disambiguation (VSD). VSD is a sub-problem of the Word Sense Disambiguation (WSD) problem, that tries to identify in which sense a polysemic word is used in a given sentence. Thus a sense inventory for each word (or lemma) must be used.
Finally, we also assess the challenges in SRL and identify the opportunities for useful further research in future.

1 Introduction

One of the fields where AI is gaining great importance is the NLP. Nowadays, NLP has many applications: search engines (semantic/topic search rather than word matching), automated speech translation, automatic summarization, etc.
Therefore, there are many sub-tasks for natural language applications that have already been studied. An example is the syntactic analysis of the words of a sentence. The object of this research study is the realization of a system able to perform SRL.
A SRL system does nothing more than take a set of input phrases and, for each of them, it starts to determine the various components that could play a semantic role. A component of a proposition that plays a semantic role is defined as constituent. Once the possible candidates are determined, Machine Learning techniques are used to label them with the right role.
This task becomes important for advanced applications where it is also necessary to process the semantic meaning of a sentence. Moreover, all this applications have to deal with ambiguity.
Ambiguity is the term used to describe the fact that a certain expression can be interpreted in more than one way.
In NLP, ambiguity is present at several stages in the processing of a text or a sentence, such as: tokenization, sentence-splitting, part-of-speech (POS) tagging, syntactic parsing and semantic processing. Semantic ambiguity is usually the last to be addressed by NLP systems, and it tends to be one of the hardest to solve among all types of ambiguities mentioned.
For this type of ambiguity, the sentence has already been parsed and, even if its syntactic analysis (parse tree) is unique and correct, some words may feature more than one meaning for the grammatical category they were tagged with.
Usually this difference in meaning is associated to syntactic properties. In order to overcome these issues, this research study approaches to the VSD task. The majority of the systems used in the VSD task are based on Machine Learning techniques (Witten, 2011).
We approach both the tasks by following two different solutions.

2 Related Work

2.1 SRL Approaches

Until recently, state-of-the-art Semantic Role Labeling (SRL) systems relied on complex sets of lexico-syntactic features (Pradhan, 2005) as well as declarative constraints (Punyakanok, 2008). Neural SRL models, instead, exploit induction capabilities of neural networks, largely eliminating the need for complex "hand-made" features. Recently, it has been shown that an accurate span-based SRL model can be constructed without relying on syntactic features (Jie Zhou, 2015). In particular, Roth and Lapata (Roth and Lapata, 2016) argue that syntactic features are necessary for the dependency-based SRL and show that performance of their model degrades dramatically if syntactic paths between arguments and predicates are not provided as an input.
Recent studies (Luheng He, 2018) propose an end-to-end approach for jointly predicting all predicates, arguments spans, and the relations between them. The model makes independent decisions about what relationship, if any, holds between every possible word-span pair, and learns contextualized span representations that provide rich, shared input features for each decision.

2.2 WSD Approaches

An overview of the most used techniques and features for WSD was also conducted, based on the systems evaluated at the SensEval3. The most common learning algorithms (Witten, 2011) used at SensEval3 are the following:

- The Naive Bayes algorithm, which estimates the most probable sense for a given word w based on the prior probability of each sense and the conditional probability for each of the features in that context.

- The Decision List algorithm (Yarowsky, 1995), which builds a list of rules, ordered from the highest to the lowest weighted feature. The correct sense of the word is determined by the first rule that is matched.

- The Vector Space Model algorithm, which considers the features of the context as binary values in a vector. In the training phase, a centroid is calculated for each possible sense of the word. These centroids are then compared with vectors of features from testing examples using the cosine function.

- Support Vector Machines, the most widely used classification technique in WSD at SensEval3 (Agirre, 2004); (Lee, 2004); (Villarejo, 2004), is a classification method that finds the maximal margin hyperplane that best separates the positive from the negative examples. In the particular case of WSD, this has to be slightly tuned for multiple class classification. Usually, methods like one-against-all are used, which lead to the creation of one classifier per class.

The most commonly used features used by the systems proposed and presented at SensEval3 can be divided as follows:

- Collocations: n-grams (usually bi-grams or tri-grams) around the target word are collected. The information stored for then-grams is composed by the lemma, word-from and part-of-speech tag of each word.

- Syntactic dependencies: syntactic dependencies are extracted among words around the target word. The relations most commonly used are subject, object, modifier. However, depending on the system, other dependencies might also be extracted.

- Surrounding context: single words in a defined window size are extracted and used in a bag-of-words approach.

- Knowledge-Based information: Some systems also make use of information suchas WordNet's domains, FrameNet's syntactic patterns or annotated examples, among others.

3 Data

The dataset used is the CoNLL 2009 Shared Task built on the CoNLL 2008 task which has been extended to multiple languages. The core of the task was to predict syntactic and semantic dependencies and their labeling.
Data was provided for both statistical training and evaluation, in order to extract these labelled dependencies from manually annotated Treebanks such as the Penn Treebank for English, the Prague Dependency Treebank for Czech and similar Treebanks for Catalan, Chinese, German, Japanese and

Spanish languages, enriched with semantic relations. Great effort has been dedicated in providing the participants with a common and relatively simple data representation for all the languages, similar to the 2008 English data. Role-annotated data makes it available for many research opportunities in SRL including a broad spectrum of probabilistic and machine learning approaches.
We have introduced the dataset associated with SRL; we are now prepared to discuss the approaches to automatic SRL and VBS.

4 Metrics

For many of these subtasks there are standard evaluations techniques and corpora. Standard evaluation metrics from information retrieval include precision, recall and a combined metric called F_1 measure (Jurafsky, 2000).
Precision is a measure of how much of the information that the system returned is correct, also known as accuracy. Recall is a measure of how much relevant information the system has extracted from text, thus a measure of the system's coverage. The F_1 measure balances recall and precision.
A corpus is often divided into three sets: training set, development set and testing set. Training set is used for training systems, whereas the development set is used to tune parameters of the learning systems, and sselecting the best model. Testing set is used for evaluation. Cross-corpora evaluation is used in some tasks, for which a fresh test set different from the training corpora is used for evaluation.
In this case, F_1 measure is computed as the harmonic mean of Precision and Recall.

5 Semantic Role Labeling

The model architecture for SRL is inspired from the one ideated by Marcheggiani et al., 2017 (Marcheggiani, 2017) based on the following three components.
Then, a table with all the hyperparameter values will be shown.

5.1 Word Representation

The word representation component builds from a word w_i in a sentence w a word representation x_i. Each word w is represented as the concatenation of four vectors:

- A randomly initialized word embedding $x_{re} \in R^{d_w}$.
- A pre-trained word embedding $x_{pe} \in R^{d_w}$.
- A randomly initialized part-of-speech tag embedding $x_{pos} \in R^{d_p}$.
- A randomly initialized lemma embedding $x_{le} \in R^{d_l}$ that is only active if the word is one of the predicates.

Then, it has been used the Predicate-Specific Encoding. Specifically, when identifying arguments of a given predicate, the authors added a predicate-specific feature to the representation of each word in the sentence by concatenating a binary flag to the word representation. The flag is set as 1 for the word corresponding to the currently considered predicate, it is set as 0 otherwise. In this way, sentences with more than one predicate will be re-encoded by Bidirectional LSTMs multiple times.

5.2 Encoder

Recurrent neural networks (RNN) (Elman, 1990), more precisely, Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are one of the most effective ways to model sequences. Formally, the LSTM is a function that takes as input the sequence and returns a hidden state. This state can be regarded as a representation of the sentence from the start to the position i, or, in other words, it encodes the word at position i along with its left context.
Bidirectional LSTMs make use oftwo LSTMs: one for the forward pass, and another for the backward pass. In this way the concatenation of forward and backward LSTM states encodes both left and right contexts of a word.
In this case, the Bidirectional Long-Short Term Memory (BiLSTM) Encoder takes as input the word representation x_i and provides a dynamic representation of the word and its context in a sentence.

5.3 Role Classifier

The goal of the classifier is to predict and label arguments for a given predicate.
The basic role classifier takes the hidden state of the top-layer bidirectional LSTM corresponding to the considered word at position i and uses it to estimate the probability of the role r.
However, since the context of a predicate in the

sentence is highly informative for deciding if a word is its argument and for choosing its semantic role, the authors provides the predicate's hidden state as another input to the classifier.
Finally, it has been proven advantageous to jointly embed the role r and predicate lemma l using a non-linear transformation: $ReLU$ (Vinod Nair and Geoffrey Hinton, 2010) that is the rectilinear activation function. In this way each role prediction is predicate-specific, and at the same time it has expected to learn a good representation for roles associated to in frequent predicates.

5.4 Hyperparameters

In the following table the hyperparameter values.

Hyperparameter	Value
English word embeddings	100
POS embeddings	16
Lemma embeddings	100
LSTM hidden states	512
Role representation	128
Output lemma representation	128
BiLSTM depth	4
Learning rate	.001

Table 1: Hyperparameter values.

6 Verb-Sense Disambiguation

In order to improve the results obtained from the Marcheggiani's SRL model, two solutions will be presented:

- Multi-Task Learning: by sharing representations between related tasks (VBS), we can enable our model to generalize better on our primary task (SRL).
- Babelfy: usage of a pre-trained model that helps to disambiguate sentences and verbs.

In the first solution the two models run in parallel. In the second solution, since the SRL model uses as input the Babelfy's output, the two models run sequentially.

6.1 Multi-Task Learning Solution

In Machine Learning we typically care about optimizing a particular metric. In order to do this, we generally train a single model to perform our desired task, then fine-tune and tweak this model until its performance no longer increases.
Even if it is possible to achieve generally acceptable performance, in this way we could miss information that might help us to optimize the relevant metric. Specifically, information deriving from the training signals of related tasks.
We can consider multi-task learning as a form of inductive transfer. Inductive transfer can help to improve a model by introducing an inductive bias, defining a model as preferable with respect to other hypotheses.
Furthermore, the Verb Sense Disambiguation model has been created; following, a brief explanation of the model. We use the same Word Representation and Encoder of the Marcheggiani's system explained in sections 5.1, 5.2.
The output of the Encoder is used to predict the sense of the verb by applying the Softmax activation function.

Model	P	R	F_1
Lei (2015)	-	-	86.6%
FitzGerald (2015)	-	-	86.7%
Roth and Lapata (2016)	88.1%	85.3%	86.7%
Marcheggiani (2017)	88.7%	86.8%	87.7%
SRL+VSD Model	88.65%	86.62%	87.6%

Table 2: Multi-Task Learning Results.

As Table 2 shows, performance worsens in terms of Precision and Recall.
Therefore, we have a lower value in term of F_1 score, which, as already mentioned above, is the harmonic mean of Precision and Recall.
For this reason another solution was developed in order to improve the results on both Precision and Recall and then of F_1.

6.2 Babelfy Solution

Babelfy (Navigli, 2014) is both a multilingual encyclopedic dictionary and a semantic network which connects concepts and named entities in a very large network of semantic relations called Babel synsets. Each Babel synset represents a given meaning and contains all the synonyms which express that meaning.
Specifically, Babelfy performs the tasks of multilingual Word Sense Disambiguation and Entity Linking.
Extracted senses have been used as input of the SRL Model, by replacing the randomly initialized

lemma embedding $x_{le} \in R^{d_l}$ of the word representation of 5.1

Model	P	R	F_1
Lei (2015)	-	-	86.6%
FitzGerald (2015)	-	-	86.7%
Roth and Lapata (2016)	88.1%	85.3%	86.7%
Marcheggiani (2017)	88.7%	86.8%	87.7%
SRL + Babelfy	88.96%	86.87%	87.9%

Table 3: Babelfy Results.

In this case we can observe improvements in all fields. This improvement is not so significant (Reimers and Gurevich, 2017) because LSTM-based models tend to be significantly sensible to initialization, for this reason 0.2% improvement in a small dataset like CoNLL2009 may not be a satisfactory increase.
Moreover, this results shows that improving the VSD task determines improvements in SRL task.

7 Conclusions

The realized work represents the development of a complete system for the Semantic Role Labeling, an important tool to be used in advanced Natural Language Processing applications.
A system of SRL alone is not very useful and it necessarily must be included in a wider application, for example a Question&Answering system or a Neural Machine Translation system.
In conclusion, as all the new applications of natural language processing must be able to handle semantic information if they want to have good performances, this type of system can be considered a valuable solution to achieve such performances.
The statistical analysis of the errors registered by the system, developing from this analysis new algorithms in order to correct such errors, is another aspect to be considered in the evaluation of this system.

8 Future Works

As for future works we could certainly try to develop a new Semantic Role Labeling model architecture trying to discover approaches related to models based on Attention.
Attention (Bahdanau, 2015) is one of the main innovations for machine translation based on neural networks, the key idea that allowed neural networks to overcome classics translation models.
The main obstacle for the sequence-to-sequence learning is the need to compress all the information contained in the original sequence into a prefixed vector. Attention alleviates this problem (Luong, 2015), allowing the decoder to look again at the list of hidden states corresponding to the original sequence, whose weighted average is used as input from the decoder in addition to the compressed vector representation.
An interesting effect of attention (Vaswani, 2017) is the possibility to observe, superficially, the operating mechanisms inside the model: the attention makes visible which parts of the input have proved important for a certain output, thanks to the weights applied to get the average of the incoming sequence.
Another future research activity could be the examination of the abovementioned models under different languages, such as Italian.

References

Witten, I. H., E. Frank, and M. A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques (3 ed.).*

Sameer Pradhan, Kadri Hacioglu, Wayne H. Ward, James H. Martin, and Daniel Jurafsky. 2005. *Semantic role chunking combining complementary syntactic views.* In Proceedings of CoNLL.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. *The importance of syntactic parsing and inference in semantic role labeling.* Computational Linguistics 34(2):257–287.

Jie Zhou and Wei Xu. 2015. *End-to-end learning of semantic role labeling using recurrent neural networks.* In Proceedings of ACL.

Michael Roth and Mirella Lapata. 2016. *Neural semantic role labeling with dependency path embeddings.* In Proceedings of ACL.

Luheng He, Kenton Lee, Omer Levy, Luke Zettlemoyer. 2018. *Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling* In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

Yarowsky D. 1995. *Unsupervised word sense disambiguation rivaling supervised methods.* In Proceedings of the 33rd annual meeting on Association for Computational Linguistics. 189–196.

Agirre E., Aldabe I., Lersundi M., Martınez D., Pociello E., Uria L. 2004. *The Basque Lexical-Sample Task* Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, Association for Computational Linguistics (2004) 1–413.

Lee, Y.K., Ng, H.T., Chia, T.K. 2004. *Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources.* Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, Association for Computational Linguistics (2004) 137–14014.

Villarejo L., Marquez L., Agirre E., Martınez D., Magnini B., Strapparava C., McCarthy D., Montoyo A., Suarez A. 2004. *The "Meaning" System on the English All-Words Task.* Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, Association for Computational Linguistics (2004) 253–256

Jurafsky D. and Martin J. H. 2000. *Machine Translation.* In Speech and Language Processing. Prentice Hall.

Diego Marcheggiani, Anton Frolov and Ivan Titov. 2017. *A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling.*

Jeffrey L. Elman. 1990. *Finding structure in time.* Cognitive Science 14(2):179–211.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. *Long short-term memory.* Neural Computation 9(8):1735–1780.

Vinod Nair and Geoffrey Hinton. 2010. *Rectified Linear Units Improve Restricted Boltzmann Machines.* ICML.

A. Moro, A. Raganato, R. Navigli. 2014. *Entity Linking meets Word Sense Disambiguation: a Unified Approach.* Transactions of the Association for Computational Linguistics (TACL), 2, pp. 231-244, 2014.

Reimers and Gurevich. 2017. *Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging.*

Bahdanau D., Cho K., and Bengio Y. 2015. *Neural Machine Translation by Jointly Learning to Align and Translate.* In ICLR 2015.

Luong M.-T., Pham H. and Manning C. D. 2015. *Effective Approaches to Attention-based Neural Machine Translation.* In Proceedings of EMNLP 2015.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Polosukhin I. 2017. *Attention Is All You Need.* In Advances in Neural Information Processing Systems.