Syntactically Aware Neural Architectures for Definition Extraction

Luis Espinosa-Anke and **Steven Schockaert**
School of Computer Science and Informatics
Cardiff University
{espinosa-ankel,schockaerts1}@cardiff.ac.uk

Abstract

Automatically identifying definitional knowledge in text corpora (Definition Extraction or DE) is an important task with direct applications in, among others, Automatic Glossary Generation, Taxonomy Learning, Question Answering and Semantic Search. It is generally cast as a binary classification problem between definitional and non-definitional sentences. In this paper we present a set of neural architectures combining Convolutional and Recurrent Neural Networks, which are further enriched by incorporating linguistic information via syntactic dependencies. Our experimental results in the task of sentence classification, on two benchmarking DE datasets (one generic, one domain-specific), show that these models obtain consistent state of the art results. Furthermore, we demonstrate that models trained on clean Wikipedia-like definitions can successfully be applied to more noisy domain-specific corpora.

1 Introduction

Dictionaries and glossaries are among the most important sources of meaning for humankind. Compiling, updating and translating them has traditionally been left mostly to domain experts and professional lexicographers. However, the last two decades have witnessed a growing interest in automating the construction of lexicographic resources.

Analogously, in Natural Language Processing (NLP), lexicographic resources have proven useful for a myriad of tasks, for example Word Sense Disambiguation (Banerjee and Pedersen, 2002; Navigli and Velardi, 2005; Agirre and Soroa, 2009; Camacho-Collados et al., 2015), Taxonomy Learning (Velardi et al., 2013; Espinosa-Anke et al., 2016b) or Information Extraction (Richardson et al., 1998; Delli Bovi et al., 2015). Moreover, lexicographic information such as definitions constitutes the cornerstone of important language resources for NLP, such as WordNet (Miller et al., 1990), BabelNet (Navigli and Ponzetto, 2012), Wikidata (Vrandečić and Krötzsch, 2014) and basically any Wikipedia-derived resource.

In this context, systems able to address the problem of *Definition Extraction* (DE), i.e., identifying definitional information spanning in free text, are of great value both for computational lexicography and for NLP. In the early days of DE, rule-based approaches leveraged linguistic cues observed in definitional data (Rebeyrolle and Tanguy, 2000; Klavans and Muresan, 2001; Malaisé et al., 2004; Saggion and Gaizauskas, 2004; Storrer and Wellinghoff, 2006). However, in order to deal with problems like language dependence and domain specificity, machine learning was incorporated in more recent contributions (Del Gaudio et al., 2013), which focused on encoding informative lexico-syntactic patterns in feature vectors (Cui et al., 2005; Fahmi and Bouma, 2006; Westerhout and Monachesi, 2007; Borg et al., 2009), both in supervised and semi-supervised settings (Reiplinger et al., 2012; Faralli and Navigli, 2013).

On the other hand, while encoding definitional information using deep learning techniques has been addressed in the past (Hill et al., 2015; Noraset et al., 2016), to the best of our knowledge no previous work has tackled the problem of DE by reconciling both the linguistic lessons learned in the past decades (e.g., the importance of lexico syntactic patterns or long-distance relations between *definiendum* and *definiens*)[1] and the processing potential of neural networks.

Thus, we propose to bridge this gap by learning high level features over candidate definitions

[1] Traditionally, a *definienidum* is a term being defined, whereas the *definiens* refers to its differentiable characteristics.

via convolutional filters, and then apply recurrent neural networks to learn long term dependencies over these feature maps. Without preprocessing and only taking pretrained embeddings as input, it is already possible to consistently obtain state of the art results in two benchmarking datasets for DE (one generic, one domain-specific). Further improvements over this simple model are obtained by incorporating syntactic information by composing and embedding head-modifier syntactic dependencies and dependency labels. One interesting side result of our experiments is the observation that a model trained only on canonical wikipedia-like definitions performs significantly better in a domain-specific academic setting than a model that has been trained on that domain, which somewhat contradicts previously assumed notions about the creativity of academic authors when presenting and describing novel terminology.[2]

2 Method

The impact of deep learning methods in NLP is today indisputable. The utilization of neural networks has improved the state of the art almost systematically in a wide number of tasks, from language modeling (Bengio et al., 2003; Yih et al., 2011; Mikolov et al., 2013) to text classification (Kim, 2014) or machine translation (Bahdanau et al., 2014), among many others.

In this paper we leverage two of the most popular architectures in deep learning for NLP with the goal to predict, given an input sentence, its probability of including definitional knowledge. In our best performing model we take advantage of Convolutional Neural Networks (CNNs) to learn local features via convolved filters (LeCun et al., 1998), and then apply to the learned feature maps a Bidirectional Long Short Term Memory (blstm) network (Hochreiter and Schmidhuber, 1997). In this way, we aim at capturing ngram-wise features (Zhou et al., 2015), which may be strong indicators of definitional patterns (e.g., the classic *X is a Y* pattern), combined with the learning of long-term sequential dependencies over these learned feature maps.

Following standard notation for sentence modeling via CNNs (Kim, 2014), we let $\mathbf{x_i} \in \mathbb{R}^k$ be the k-dimensional word vector associated to the i-th word in an input sentence $\mathbf{S}$. We use as pre-trained embeddings the *word2vec* (Mikolov et al., 2013) vectors trained with negative sampling on the Google News corpus[3]. Each sentence is represented as an $n \times k$ matrix $\mathcal{S}$, where n is the size of the longest sentence in the corpus (using padding where necessary). The convolution layer applies a filter $\mathbf{w_j} \in \mathbb{R}^{(h+1)k}$ to each ngram window of $h+1$ tokens. Specifically, writing $\mathbf{x_{i:i+h}}$ for the concatenation of the word vectors $\mathbf{x_i}, \mathbf{x_{i+1}}, ..., \mathbf{x_{i+h}}$, we have:

$$c_j^i = f\left(\mathbf{w_j} \cdot \mathbf{x_{i:i+h}} + b_j\right)$$

where $b_j \in \mathbb{R}$ is a bias term and f is the ReLu activation function (Nair and Hinton, 2010). In total, we use 100 such convolutional features, i.e. we use the vector $\mathbf{c^i} = \left[c_1^i, c_2^i, \cdots, c_{100}^i\right]$ to encode the i^{th} ngram. We empirically set the length $h+1$ of each ngram to 3. To reduce the size of the representation, we then use a max pooling layer with a pool size of 4. Let us write $\mathbf{d^i} = \left[d_1^i, d_2^i, \cdots, d_{97}^i\right]$, where $d_j^i = \max(d_j^i, d_j^{i+1}, d_j^{i+2}, d_j^{i+3})$. The input sentence S is then represented as the sequence $\mathbf{d^1}, \mathbf{d^5}, \mathbf{d^9}, ..., \mathbf{d^{n-3}}$, which is used as the input to a bidirectional LSTM (BLSTM) layer. Finally, the output vectors of the final states for both directions of this BLSTM are connected to a single neuron with a sigmoid activation function. In all the experiments reported in this paper, we classify a sentence as definitional when the output of this neuron yields a value which is at least 0.5.

2.1 Incorporating Syntactic Information

The role of syntax has been extensively studied for improving semantic modeling of domain terminologies. Examples where syntactic cues are leveraged include medical acronym expansion (Pustejovsky et al., 2001), hyponym-hypernym extraction and detection (Hearst, 1992; Shwartz et al., 2016), and definition extraction either from the web (Saggion and Gaizauskas, 2004), scholarly articles (Reiplinger et al., 2012), and more recently from Wikipedia-like definitions (Boella et al., 2014).

However, the interplay between syntactic information and the generalization potential of neural networks remains unexplored in definition modeling, although intuitively it seems reasonable to assume that a syntax-informed architecture should have more tools at its disposal for discriminating

[2] Code available at bitbucket.org/luisespinosa/neural_de

[3] code.google.com/archive/p/word2vec/

between definitional and non-definitional knowledge. As an example of the importance of syntax in encyclopedic definitions, among the definitions contained in the WCL definition corpus (see Section 3.1), 71% of them include the lexico-syntactic pattern $\texttt{noun} \xleftarrow{\text{subj}} \texttt{is} \xrightarrow{\text{dobj}} \texttt{noun}$. To explore the potential of syntactic information, we represent dependency-based phrases by embedding them in the same vector space as the pretrained word embeddings introduced above. This approach draws from previous work on modeling phrases by composing their parts and the relations that link them (Socher et al., 2011, 2013, 2014).

Specifically, let $\mathcal{S}_d$ be the list of head-modifier relations obtained by parsing[4] sentence $\mathbf{S}$. Each relation r in $\mathcal{S}_d$ is a head-modifier tuple $\langle h, m, l \rangle$. Here l denotes the dependency label of the relation (e.g., *nsubj*), which we represent as the vector $\mathbf{r} = \frac{1}{2}(\mathbf{h} + \mathbf{m})$, with $\mathbf{h}$ and $\mathbf{m}$ the vector representations of words h and m respectively. This setting for composing first-order head-modifier relations is similar to the one proposed in Dyer et al. (2015) for dependency parsing. This leads to a represention of the sentence as a sequence $\mathbf{r_1}, ..., \mathbf{r}_{|\mathcal{S}_\mathbf{d}|}$, which preserves the original order of head words. The intuition is that this "coarser" grained sorting[5] provides integrated semantic-syntactic information that can be leveraged both by the convolutional feature extraction step, and more importantly, by the sequential BLSTM module.

Then, for each sentence we concatenate the dependency-based representation $\mathbf{r_1}, ..., \mathbf{r}_{|\mathcal{S}_\mathbf{d}|}$ to the word vector sequence $\mathbf{x_1}, ..., \mathbf{x_n}$, to obtain the input to the convolutional layer of our model. It is worth mentioning that we tried different merging schemes (concatenation, but also dot product and averaging) at different layers, and found that the best way to inform our neural definition extractor is to encode this syntactic information explicitly at input time. Finally, we also explore the effect of enriching the input representation with the information of the dependency label. For each sentence, we enrich each head-modifier mean vector $\mathbf{r_i}$ by concatenating them a one-hot representation of their corresponding dependency label. The search space of these labels is 46 (e.g., *nsubj* or *dobj*). An illustrative diagram of our proposed architecture is provided in Figure 1.

[4]We use the dependency parser provided in the SpaCy NLP library: `spacy.io`.

[5]It is coarser because in a dependency tree modifiers naturally lose their original order.

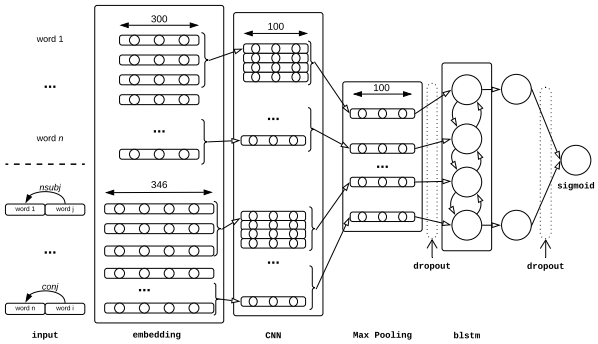

Figure 1: Architecture of our proposed definition extraction model. Input may be either simple pretrained embeddings or syntactically enriched representations (separated by the dotted line).

3 Evaluation

3.1 Evaluation data

WCL: The WCL (World-Class Lattices) dataset (Navigli et al., 2010) consists of manually annotated Wikipedia definitions and distractors (1,871 and 2,847 respectively). These distractors are sentences that also include the term (i.e., the Wikpedia page title) and are what the authors call "syntactically plausible false definitions". The style of the definitions is fairly consistent, and follows in most cases the Aristotelian *genus et differentia* structure of a definition (A is a B which C). We list below both an example definition and one of its distractors:

✓ *The **Amiga** is a family of personal computers originally developed by Amiga Corporation.*

✗ *Development on the **Amiga** began in 1982 with Jay Miner as the principal hardware designer.*

W00: Introduced in Jin et al. (2013), this corpus consists of a collection of 731 definition sentences compiled from the ACL-ARC anthology (Bird et al., 2008), and 1454 distractors. Their style is different[6], as they are used mostly for introducing and describing novel terminology in NLP research papers. Let us show an example for each sentence class:

[6]In lexicographic terms, most definitions in this dataset would be classified either as *extensional* (definition without hypernym) or *functional* (define something by *what it does*. instead of what it is).

✓ *Our system, **SNS** (pronounced "essence"), retrieves documents related to an unrestricted user query and summarizes a subset of them as selected by the user.*

✗ *The **senses** with the highest confidence scores are the senses that contribute the most to the maximization function for the set .*

3.2 Baselines

Let us provide a succint description of each competing baseline. **(1) WCL**: An algorithm that learns word-class lattices for modeling higher-level features over shallow parsing and part of speech (Navigli and Velardi, 2010). **(2) DefMiner**: A CRF-based sequential modeling system trained with lexical, terminological and structural (e.g., document position) features (Jin et al., 2013). **(3) B&DC**: A binary classifier trained with dependency paths over input sentences (Boella et al., 2014). **(4) E&S**: A system based on more complex dependency-based features (Espinosa-Anke and Saggion, 2014). **(5) LSTM-POS**: An LSTM-based system which represents each sentence as a mixture of infrequent words and frequent words' associated part-of-speech (Li et al., 2016).

As for our proposed models, we include results for a CNN architecture alone (**CNN**), as well as for the proposed CNN and BLSTM (**C-BLSTM**) combination. For both architectures, subscripts d or l denote the syntactically informed variant without and with one-hot label encoding information, respectively. Finally, among the many hyperparameters that can be explored, we report the impact of the dimensionality of the output vectors of the BLSTM layer, with sizes of 100 and 300. We did not attempt to tune the other hyperparameters.

Experiment 1: In-domain 10-fold CV

In this experiment, we compare the performance of different configurations of our proposed model with previous contributions in a 10-fold cross validation (CV) setting. The experimental results, listed in Table 1, show that a fairly simple CNN architecture with no preprocessing already achieves remarkably strong results, especially for the WCL dataset. Among our proposed systems, the overall best performance in Wikipedia definitions is obtained by the CNN$_l$ configuration. However, incorporating a BLSTM layer contributes towards the best performing model on the NLP-specific dataset (C-BLSTM100$_d$). Several conclusions can be drawn from these results. First, CNNs are capable of capturing a great deal of Wikipedia-like definitional information. This probably owes to the fairly recurrent linguistic structure of these definitions. On the contrary, however, LSTMs seem necessary in more complex scenarios, e.g., in those presented in the W00 dataset. Here, we argue that long term dependencies may play an important role, for example, for capturing cases where a full-fledged definitions appear spanning only over the last tokens of a sentence. Finally, syntax seems to help for most configurations, and for both datasets, although the difference is more pronounced in the more challenging W00 dataset.

	WCL			W00		
	P	R	F1	P	R	F1
WCL	**98.8**	60.7	75.2	-	-	-
DefMiner	92.0	79.0	85.0	-	-	-
B&DC	88.0	76.0	81.6	-	-	-
E&S	85.9	85.3	85.4	-	-	-
LSTM-POS	90.4	92.0	91.2	-	-	-
CNN	91.1	92.0	91.5	33.5	68.7	44.8
CNN$_d$	90.6	90.9	90.7	34.2	**69.4**	45.8
CNN$_l$	94.2	**94.2**	**94.2**	42.8	65.5	51.3
C-BLSTM100	93.3	91.8	92.5	46.1	68.7	54.5
C-BLSTM100$_d$	93.2	92.2	92.6	**52.0**	67.6	**57.4**
C-BLSTM100$_l$	93.2	92.7	92.9	51.7	66.2	57.3
C-BLSTM300	93.4	92.3	92.7	48.9	64.5	54.0
C-BLSTM300$_d$	94.3	91.0	92.6	47.3	64.0	51.9
C-BLSTM300$_l$	94.0	90.7	92.5	50.0	64.5	53.8

Table 1: Comparative results between previous contributions and different configurations of our proposed contribution.

These differences in performance are, however, small enough to make it difficult to draw strong conclusions other than that neural network architectures are a sensible choice for this task, and that syntax can play an important role depending on the type of data to be processed. It is important to highlight, finally, that depending on the application, one may be more interested in having an almost perfect precision (as in the system described in Navigli and Velardi (2010)). For automatic glossary generation from text, on the other hand, having a more balanced model, with high re-

call at the expense of only slightly lower precision, may be preferred, as automatic glossaries usually undergo a human post-editing and revision step.

Experiment 2: Cross-domain DE

In this experiment we assess the performance of a cross-domain model on the W00 dataset (cf. Section 3.1). The main goal is to verify to what extent a model trained only on Wikipedia-like definitions can do well in a domain-specific setting. To this end, we apply our best performing configuration trained on the whole WCL corpus to the W00 dataset (**WCL>W00**), and compare it with the performance of our best configuration as per 10-fold CV (**C-BLSTM100**$_d$, see Table 1). This experiment is important, for example, for learning what would be more appropriate if we were to aim at constructing domain-specific glossaries or at extracting highly specific semantic relations from a domain terminology.

System	Precision	Recall	F-Score
C-BLSTM100$_d$	52.0	67.6	57.4
WCL>W00	**69.0**	**71.0**	**70.0**

Table 2: Results of our proposed model (with two different training schemes) on the NLP-specific W00 definition dataset.

The results in Table 2 reveal that, despite differences in style, a system modeled over encyclopedic definitions outperforms a neural model trained only on these idiosyncratic definitions. This might be due to several reasons. First, because of the slightly smaller size of this dataset. And second, the more noisy nature of the corpus may pose a stronger challenge for a neural model to identify recurrent definitional patterns. Still, our experimental results seem to suggest that these patterns do exist, as evidenced by the strong performance of the Wikipedia-trained model.

Qualitative Evaluation

We run our best performing model over a subset of the ACL-ARC anthology (Bird et al., 2008), specifically the subcorpus described in (Espinosa-Anke et al., 2016a), which removed noisy sentences as produced by the pdf to text conversion.

In Table 3 we show three high quality definitions discovered by our model, as well as three false positives. We may highlight the somewhat surprising remarkable capacity of the model to identify definitions beyond the is-a pattern (e.g., using the verb 'mean') and with long-distance dependencies between subject and object. As for the incorrect cases, we find that for this model to be used in the automatic glossary construction task, in addition to further refinement, it would have to be coupled with a term extraction system so that only definitions associated to meaningful domain terms are extracted.

`compositional grammar means that the semantics of a a phrase is composed of the semantics of the subphrases`
`f-score is the harmonic mean of recall (r) and precision (p) percentages`
`silc is a language and encoding identification system developed by the rali laboratory at the university of montreal`
`the main lesson is that complex sentences are analysed with a proper understanding without sacrificing efficiency`
`a simple spell correction is a part of the system (essentially 1 character errors)`
`the segmentation of a translation memory is a key feature for our system`

Table 3: Examples of extracted definitions with over 0.9 confidence from a subset of the ACL-ARC corpus.

4 Conclusion

We have presented and evaluated a neural model based on CNNs and Bidirectional LSTMs which obtains state of the art results on two well known *definition extraction* datasets. From our experiments, it stems that: (1) Neural network architectures perform well for identifying definitional text snippets in corpora, more so with syntactic information; (2) A model trained on Wikipedia is competitive even in a domain-specific setting; and (3) More complex linguistic structures seem to be better captured with more complex models. As for future work, it would be interesting to explore whether meaningful further gains can be obtained by performing hyperparameter tuning.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. We would also like to thank Miguel Ballesteros for fruitful discussions. This work was supported by ERC Starting Grant 637277.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 33–41.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pages 136–145.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08)*. European Language Resources Association (ELRA), Marrakech, Morocco. ACL Anthology Identifier: L08-1005.

Guido Boella, Luigi Di Caro, Alice Ruggeri, and Livio Robaldo. 2014. Learning from syntax generalizations for automatic semantic annotation. *Journal of Intelligent Information Systems* pages 1–16.

Claudia Borg, Michael Rosner, and Gordon Pace. 2009. Evolutionary algorithms for definition extraction. In *Proceedings of the 1st Workshop in Definition Extraction*.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A unified multilingual semantic representation of concepts. In *ACL (1)*. pages 741–751.

Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2005. Generic soft pattern models for definitional question answering. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 384–391.

Rosa Del Gaudio, Gustavo Batista, and António Branco. 2013. Coping with highly imbalanced datasets: A case study with definition extraction in a multilingual setting. *Natural Language Engineering* pages 1–33.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *TACL* 3:529–543.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075* .

Luis Espinosa-Anke, Roberto Carlini, Horacio Saggion, and Francesco Ronzano. 2016a. Defext: A semi supervised definition extraction tool. In *Globalex,*.

Luis Espinosa-Anke and Horacio Saggion. 2014. Applying dependency relations to definition extraction. In *Natural Language Processing and Information Systems*, Springer, pages 63–74.

Luis Espinosa-Anke, Horacio Saggion, Francesco Ronzano, and Roberto Navigli. 2016b. Extasem! extending, taxonomizing and semantifying domain terminologies. In *Proceedings of the 30th Conference on Artificial Intelligence (AAAI16)*.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*.

Stefano Faralli and Roberto Navigli. 2013. Growing multi-domain glossaries from a few seeds using probabilistic topic models. In *EMNLP*. pages 170–181.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*. pages 539–545.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2015. Learning to understand phrases by embedding the dictionary. *arXiv preprint arXiv:1504.00548* .

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

Yiping Jin, Min-Yen Kan, Jun-Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the ACL anthology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 780–790.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Judith Klavans and Smaranda Muresan. 2001. Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proceedings of the AMIA*

Symposium. American Medical Informatics Association, page 324.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

SiLiang Li, Bin Xu, and Tong Lee Chung. 2016. Definition extraction with lstm recurrent neural networks. In *China National Conference on Chinese Computational Linguistics*. Springer, pages 177–189.

Véronique Malaisé, Pierre Zweigenbaum, and Bruno Bachimont. 2004. Detecting semantic relations between terms in definitions. In *CompuTerm 2004 - 3rd International Workshop on Computational Terminology*. pages 55–62.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*. pages 746–751.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography* 3(4):235–244.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. pages 807–814.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

Roberto Navigli and Paola Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on pattern analysis and machine intelligence* 27(7):1075–1086.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *ACL*. pages 1318–1327.

Roberto Navigli, Paola Velardi, and Juana María Ruiz-Martínez. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of LREC'10*. Valletta, Malta.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2016. Definition modeling: Learning to define word embeddings in natural language. *arXiv preprint arXiv:1612.00394* .

James Pustejovsky, J Castano, Jason Zhang, M Kotecki, and B Cochran. 2001. Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proceedings of the Pacific symposium on biocomputing*. volume 7, pages 362–373.

Josette Rebeyrolle and Ludovic Tanguy. 2000. Repérage automatique de structures linguistiques en corpus : le cas des énoncés définitoires. *Cahiers de Grammaire* 25:153–174.

Melanie Reiplinger, Ulrich Schäfer, and Magdalena Wolska. 2012. Extracting glossary sentences from scholarly articles: A comparative evaluation of pattern bootstrapping and deep analysis. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, Jeju Island, Korea, pages 55–65.

Stephen D Richardson, William B Dolan, and Lucy Vanderwende. 1998. Mindnet: acquiring and structuring semantic information from text. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics, pages 1098–1102.

Horacio Saggion and Robert Gaizauskas. 2004. Mining on-line sources for definition knowledge. In *17th FLAIRS*. Miami Bearch, Florida.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*. pages 801–809.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics* 2:207–218.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in German text corpora. In *Conference on Language Resources and Evaluation (LREC)*.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39(3):665–707.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM* 57(10):78–85.

Eline Westerhout and Paola Monachesi. 2007. Combining pattern-based and machine learning methods to detect definitions for elearning purposes. In *Proceedings of RANLP 2007 Workshop Natural Language Processing and Knowledge Representation for eLearning Environments*.

Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 247–256.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630* .