# Detecting Multi Word Terms in Patents the same way as Named Entities

Tobias Fink[a], Linda Andersson[b] and Allan Hanbury[c]

[a]TU Wien, tobias. ink@tuwien.ac.at
[b]TU Wien and Arti icial Researcher IT GmbH, linda.andersson@tuwien.ac.at [c]TU Wien, allan.hanbury@tuwien.ac.at

ABSTRACT

In English patent document information retrieval, Multi Word Terms (MWTs) are an important factor in determining how relevant a patent document is for a particular search query. Detecting the correct boundaries for these MWTs is no trivial task and often complicated by the special writing style of the patent domain. In this paper we describe a method for detecting MWTs in patent sentences based on a method for detecting technical named entities using deep learning. On our annotated dataset of 22 patents, our method achieved an average precision of 0.75, an average recall of 0.74 and an average F1 score of 0.74. Further, we argue for the use of domain specific word embedding resources and suggest that our model mostly learns whether individual words should be included in MWTs or not.

*Keywords*: deep learning, multi word term, patent IR, named entity recognition

## 1. Introduction

Domain specific terminology and technical language often play a key role when determining whether a particular patent document is relevant for a particular search query in Patent Information Retrieval (IR). In English, technical terms of this domain specific terminology are often composed of multiple words making them Multi Word Terms (MWTs), such as "blood cell count". The meaning of a MWT can be different from the combined meaning of the individual words, which makes it important to detect MWTs as units. When identifying MWTs important words that contribute to the technical nature of the term need to be included and non-technical words need to be excluded. Whether an individual word is an important part of the MWT is not always obvious to the non-expert and might depend on the context of the patent. For example, a "shiny appearance" can be a necessary piece of information in the context of baking products but might be a subjective addition by the author in any other context (see [4]). New MWTs are frequently introduced in the patent domain, be it because of new technology / new concepts that need new MWTs to describe them or be it because of paraphrasing of existing concepts so that the used MWTs refer to a concept more abstractly to widen the scope of a patent claim [6]. As a result, some MWTs that define key-concepts of a technology do not occur very frequently in a patent corpus. In this paper we present a method for detecting MWTs in patent sentences inspired by deep learning methods for detecting keyphrase named entities in scientific text (See [1]). We compare the performance of various model components using a dataset of 22 patents with annotated MWTs. Further, we provide a qualitative analysis of the model performance by looking at the non-training data prediction errors.

## 2. Multi Word Term Extraction

Since technical terms are often Noun Phrases (NP) ([5]), many methods (such as [3]) require Part-of-Speech (PoS) tagging to detect MWTs. However, [2] note that due to the unique writing style in the patent domain the quality of PoS tagging patent text is problematic, which is why we opt to use a method that does not require PoS tagging to work.
We conduct our experiments on a small dataset of 22 patent documents randomly selected from the CLEF-IP 2013 Topic patent document set. For this dataset we manually annotate the MWT boundaries (i.e. the MWT start and end indices) as they appear in the plain text patent document. Sentences are split into word-token sequences and each word is also split into a sequence of 32 characters. In total, our dataset consists of 232,065 word tokens, 10,337 sentences, and 19,465 MWT instances from a dictionary of 5,099 MWTs. The average MWT dictionary size per patent is 241, while the standard deviation of the MWT dictionary size is 335.
Following the method described in [1], we create a MWT-model architecture (Figure 1) that is designed to transform an input sentence represented as a sequence of words into a BILOU encoded output sequence of labels representing the MWTs in the sentence. The architecture consists of the following components:

- **Word Embedding Component:** consists of a pre-trained word vector which is concatenated to a character representation produced by a small Character-CNN component. We compare domain specific word embeddings with general purpose word embeddings as well as the impact of character representations.
- **LSTM Component:** consists of two Bi-directional LSTM layers.
- **Scoring Component:** produces a sequence of label score vectors, containing a score for each BILOU label.
- **CRF Component:** takes the sequence of label score vectors and predicts the most likely label sequence.

The predicted label sequence is converted to a prediction of MWT boundaries, which are then compared to the



*Published in "Proceedings of The 1**st Workshop on Patent Text Mining and Semantic Technologies (PatentSemTech 2019)**", Eds. L. Andersson, H. Aras, F. Piroi, A. Hanbury, PatentSemTech 2019, 12 September 2019, Karlsruhe, Germany. https://doi.org/10.34726/pst2019.3|* 

*An extended version has been submitted to the World Patent Information Journal - Virtual Special Issue: Text Mining and Semantic Technologies in the Intellectual Property domain.*

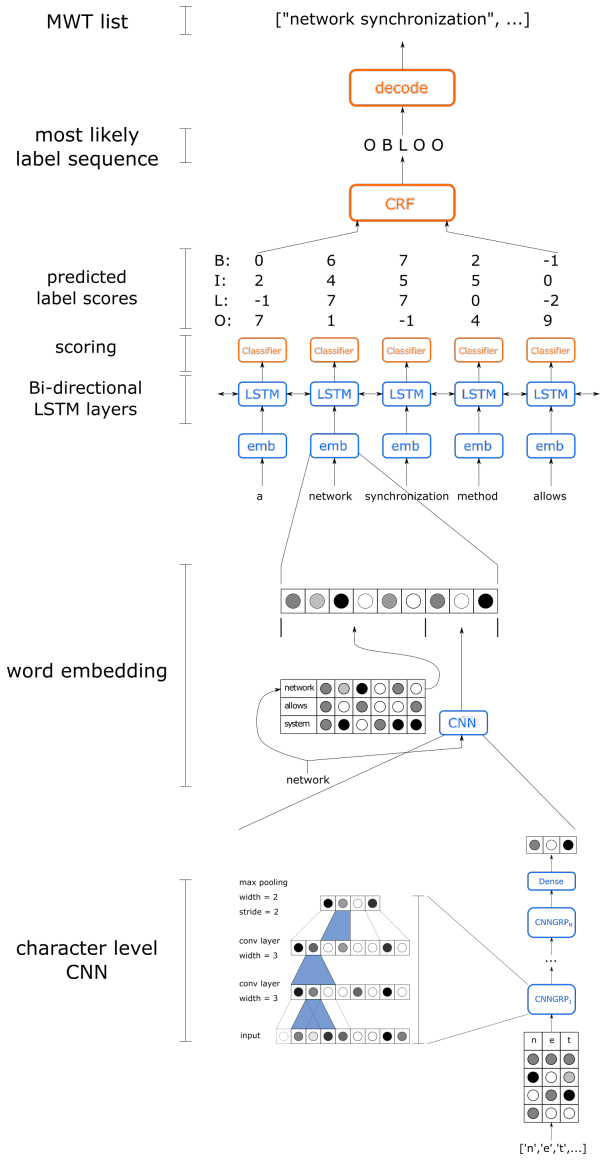


**Figure 1:** The complete architecture of the MWT model.

ground truth MWT boundaries. To prevent model overfitting we employ early stopping: we keep 10% of our training set patents as validation set and stop training if the validation set F1 score does not improve for a set number of epochs. To measure the performance when detecting MWTs in patent texts, we calculate the precision, recall and F1 score of the model predictions. A MWT prediction counts as a True Positive only if the start and end boundaries exactly match the ground truth boundaries. Further, we provide a qualitative analysis of the model's performance, in particular with respect to prediction errors and their possible causes.

## 3. Results

Our experiments show that using word embeddings pre-trained on the patent domain outperforms the use of word embeddings pre-trained on Wikipedia and results in an average precision of 0.75, an average recall of 0.74 and an average F1 score of 0.74. In fact, it is necessary to use domain specific word embeddings paired with a character representation produced by the Character-CNN component to perform better than a simple Noun Phrase filter that just annotates all Noun Phrases as MWTs.

Further, we investigate the errors that are made during prediction to get a better idea how the model could be improved. Going through the sentences and the predictions of our best model revealed that the model misses some MWTs by leaving out some words that should be attributed a technical nature, such as "distributed". This out-of-vocabulary problem might be the result of a too small training set. Sometimes, the model also adds words to MWTs that should not be included, such as non-technical words containing the sub-strings 'activ' and 'ing'. However, these sub-strings also frequently appear in words that are part of true MWTs, which explains the model's behaviour.

## 4. Conclusion

Our experiments suggest that a small dataset of only 22 patents results in an out-of-vocabulary problem and that both a patent specific word embedding resource as well as character representations of words are needed to perform better than basic NP-Filtering. The network appears to learn whether or not individual words or character sequences should be attributed a technical nature, adding them to MWTs if they appear in a MWT context during training or leaving them out if they do not. The same word being included in one MWT but excluded from other MWTs was almost never observed.

By increasing the dataset size it might be possible to reduce the out-of-vocabulary problem in future work. Furthermore, adding additional components, such as a gazetteer or pre-trained language model component, might also improve model performance.

## References

[1] Ammar, W., Peters, M., Power, R., Bhagavatula, C., 2017. The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. nucleus 2, e2. URL: https://pdfs.semanticscholar.org/2264/e14e35dc5a3db93437bc408a03171af8c59d.pdf.

[2] Andersson, L., Lupu, M., Palotti, J., Hanbury, A., Rauber, A., 2016. When is the Time Ripe for Natural Language Processing for Patent Passage Retrieval?, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM. pp. 1453–1462. URL: http://dl.acm.org/citation.cfm?id=2983858.

[3] Anick, P.G., Verhagen, M., Pustejovsky, J., 2014. Identification of Technology Terms in Patents., in: LREC, pp. 2008–2014. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/701_Paper.pdf.

[4] van Dulken, S., 2014. Do you know English? The challenge of the English language for patent searchers. World Patent Information 39, 35–40.

[5] Justeson, J.S., Katz, S.M., 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. Natural language engineering 1, 9–27.

[6] Nanba, H., Kamaya, H., Takezawa, T., Okumura, M., Shinmori, A., Tanigawa, H., 2009. Automatic translation of scholarly terms into patent terms, in: Proceedings of the 2nd international workshop on Patent information retrieval, ACM. pp. 21–24.