User Interface for Managing and Refining Related Patent Terms

Girish Showkatramani, Arthi Krishna(✉), Ye Jin, Aaron Pepe, Naresh Nula, and Greg Gabel

United States Patent and Trademark Office, Alexandria, VA, USA
{Girish.Showkatramani, Arthi.Krishna, Ye.Jin, Aaron.Pepe, Naresh.Nula, Greg.Gabel}@uspto.gov

Abstract. One of the crucial aspects of the patent examination process is assessing the patentability of an invention by performing extensive keyword-based searches to identify related existing inventions (or lack thereof). The expertise of identifying the most effective keywords is a critical skill and time-intensive step in the examination process. Recently, word embedding [1] techniques have demonstrated value in identifying related words. In word embedding, the vector representation of an individual word is computed based on its context, and so words with similar meaning exhibit similar vector representation. Using a number of alternate data sources and word embedding techniques we are able to generate a variety of word embedding models. For example, we initially clustered patent data based on the different areas of interests such as Computer Architecture or Biology, and used this data to train Word2Vec [2] and fastText [3] models. Even though the generated word embedding models were reliable and scalable, none of the models by itself was sophisticated enough to match an experts choice of keywords.

In this study, we have developed a user interface (Fig. 1) that allows domain experts to quickly evaluate several word embedding models and curate a more sophisticated set of related patent terms by combining results from several models or in some cases even augmenting to them by hand. Our application thereby seeks to provide a functional and usable centralized interface towards searching and identifying related terms in the patent domain.

Keywords: Human-computer interaction · Natural language processing
Patent · Synonyms · Word embedding · Patent search · Word similarity
Clustering

1 Introduction

Searching prior art to determine whether or not a patent can be granted for a claimed invention is one of the most crucial and time intensive aspect of the patent examination. This is owing to the fact that a comprehensive prior art search is involved in deciding whether any previous publication discloses the claimed invention or not. Currently, patent examiners perform extensive keyword based searches in existing patents and other published documents (non-patent literature) using a variety of sources such as technical documents, computer databases etc. to identify terms that are related to the

C. Stephanidis (Ed.): HCII Posters 2018, CCIS 850, pp. 115–120, 2018.
https://doi.org/10.1007/978-3-319-92270-6_16

filed patent application. This process of manually searching and evaluating results is very laborious and time consuming and the length of search time depends on the complexity of the invention. Furthermore, the overall efficiency in identifying related patent terms in existing patents and publications heavily relies on the expertise of the patent examiner.

In recent years, word embeddings have been shown to provide representation of words in a meaningful way and thus have become popular in Natural Language Processing applications [4–6]. In word embedding techniques, each word is represented as a vector in an n-dimension space, learned from very large dataset, in a way such that the semantic relationship between the words is captured. Mostly, word embedding techniques are based on Harris distributional hypothesis [7], which states that words that occur in the same contexts tend to have similar meanings. Individual words are represented as vectors of real numbers using their context such that similar words tend to have similar vector representations. Furthermore, over the years, numerous algorithms have been developed to generate word embedding models [8–12]. More recently, shallow neural networks based methods that can learn phrases vector representations have been introduced and have found to be very effective on word similarity and analogy tasks [2, 3].

The purpose of this study is to develop a centralized user interface that can be help patent examiners identify related patent terms and thus aid in making decisions regarding prior art.

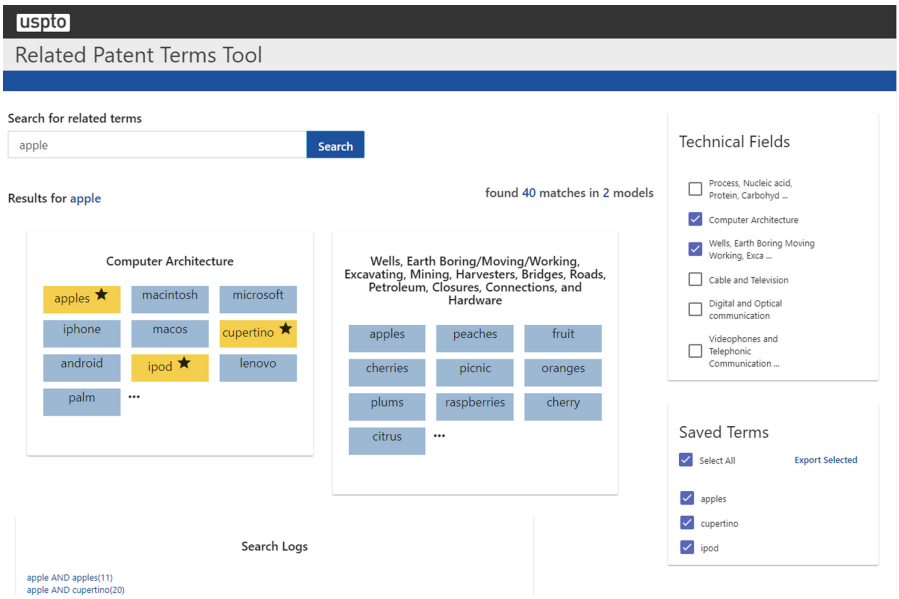

Fig. 1. Related patent terms.

2 Related Terms Search

We have built a search system that will allow patent examiners and other domain experts to evaluate the quality of related patent terms returned by the different models. In our study, we initially created clusters of patent data depending on the areas of interests such as Computer Architecture, Biology, Surface Transportation, Plants, Optics etc. to name a few. These clustered groups of patent data were then trained to generate word embedding models. The word embedding models were then utilized to query and identify related patent terms in the patent corpus. For example, when "*apple*" is entered as an input key term, all the terms related to apple are displayed in the Computer Architecture group and the oil wells group (see Fig. 1). Our application thereby seeks to provide a centralized interface that will act as a knowledge management and curation tool to patent examiners in identifying related patent terms.

The "Technical Fields" menu allows users to choose the different models to compare. For each chosen model, the corresponding tile with top 10 closest words appears in the results. Clicking on the ellipsis expands the result set can and shows more terms. User is able to choose, by clicking on the star icons, multiple terms across several models. As shown in "Saved Terms" menu on the right side, these selected terms can then be exported. The following sections describe the data set variations and alternate word embedding technologies used to create these models.

3 Dataset

The patent text for the granted United States patents and pre grant published patent applications, available publically on the (http://patents.reedtech.com/), serves as the primary data source. All of the patent text including the full text describing the invention and tables, mathematical expressions etc. are used to train the word embedding models.

The patent dataset is first clustered in to different areas of interests such as Computer Architecture, Biology, Surface Transportation, Plants, Optics etc. This clustered raw data is pre-processed, for example, stop words and non-alphanumeric characters are removed. The processed text acts as input for generating word embedding models (see Fig. 2).

4 Methods

Two state-of-the-art methods are used for generating word embeddings: Word2Vec [2] and fastText [3].

4.1 Word2Vec

The processed clustered data is trained using the Word2Vec neural network model. Word2vec is a group of related models that learn word embeddings in an unsupervised manner. These models are shallow neural networks that are trained to capture the

semantic relationship between words. Word2vec models take large corpus of text as input and encode each unique tokens as N dimensional vectors. The mapping of words to vectors allows words with similar context to be placed proximally closer to one another in the vector space. The following hyperparameters were set for training the skip-gram model: minimum count of words, size of negative samples, embedding size and epochs to train.

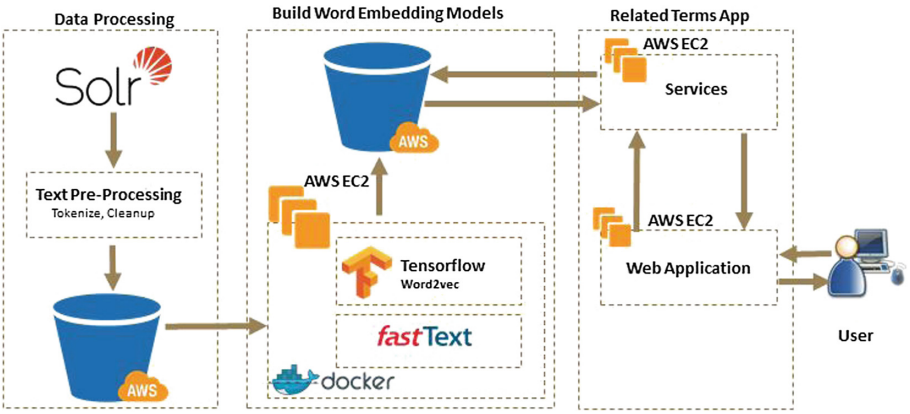

Fig. 2. System architecture.

4.2 fastText

We also trained the processed clustered data using fastText word embedding method. In the process of computing embedding, fastText enhances word vectors with subword information using additional character N-gram of variable length. This allows the algorithm to identify prefixes, suffixes, morphological nuances and syntactic structure and makes them useful for morphologically rich languages. The following hyperparameters were set for training the fastText word vector model: window size, learning rate, wordNgrams, minimum count of words, dimensions and epochs to train.

5 Infrastructure

Amazon Web Services (AWS) cloud infrastructure and Docker [13] were utilized for identifying related patent terms. AWS m4.16xlarge (64 Core Intel Xeon E5-2676 v3 Haswell processor and 256 GB DDR3 RAM) spot instance was utilized for generating word embeddings. AWS EC2 spot instances were chosen since they have an advantage of providing surplus of computing resource at a lower price compared to the on-demand instance price. In addition, Docker light-weight containers were configured to ease the configuration and setup of the TensorFlow framework [14].

6 Explanation – Search Logs

Word embedding models are flexible and a general way to capture co-occurrence relations between words. However, these models do not offer explanations or reasons for the answers. Often when users spot an unexpected word in the related terms, the users often want to know why. Our use case, patent search, lends itself to explaining the occurrence. By mining patent search logs shared by examiners, we are able to show instances of patent examiners having used the two terms together in a search query. These examples offer insights into why and how the related term is used (Fig. 3).

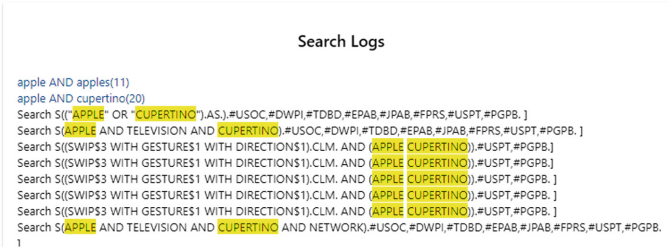

Fig. 3. Explanation from search logs.

7 Conclusion

Using recent advances in word embedding [1] techniques we have been able to produce several models for identifying patent terms related to a keyword. However, none of the models by itself was sophisticated enough to match an experts choice of keyword expansion. In this effort, we have designed and implemented a centralized user interface that allows users to consolidate and curate related terms across different models. In addition, the interface offers explanation and insight into why the related term was suggested. In future work, we would like to explore crowd sourcing of the curated patent terms.

Acknowledgments. The authors would like to thank David Chiles, David Landrith and Thomas Beach for their support of this effort.

References

1. Yitan, L., Linli, X.: Word embedding revisited: a new representation learning and explicit matrix factorization perspective (PDF). In: International Joint Conference on Artificial Intelligence (2015)
2. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representation of words and phrases and their compositionality. In: NIPS: Proceedings of Neural Information Processing Systems Nevada, USA, pp. 3111–3119 (2013)
3. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification (2016)

4. Bojanowski, P., Joulin, A., Mikolov, T.: Alternative structures for character-level RNNs (2015)
5. Zhang, X., Zhao, J., Lecun, Y.: Character-level convolutional networks for text classification. In: Neural Information Processing Systems (2015)
6. Yoon, K., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models (2015)
7. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
8. Harris, Z.: Distributional structure. Word **10**(23), 146–162 (1954)
9. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**, 141–188 (2010)
10. Baroni, M., Lenci, A.: Distributional memory: a general framework for corpus-based semantics. Comput. Linguist. **36**(4), 673–721 (2010)
11. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track, Arizona, USA (2013). arxiv.org/abs/1301.3781
13. Merkel, D.: Docker: lightweight linux containers for consistent development and deployment (2014)
14. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015)