translation point of view were identified, as well as stylistic techniques for the translation of phraseological units were identified. The choice of a particular type of translation depends on the features of the phraseological units that the interpreter must recognize and be able to convey their meaning, brightness and expressiveness. Phraseological translation involves the use in the text of the translation of stable units of varying proximity degrees between the unit of the foreign language and the corresponding unit of the translation language – from the full and absolute equivalent to the approximate phraseological correspondence.

Conclusion. Adverbial phraseological units in both English and Ukrainian are characterized by the complexity of semantics and express a wide range of concepts. We have also found out that according to A.V. Kunin's conclusion, adverbial phraseological units from the point of view of their semantic features are divided into two classes: qualitative and circumstantial. Moreover, we can conclude that the main thing in the translation of phraseological units is that one can never substitute the color of the idiomatic colors of the target language, that is it is better to refuse at all to transfer the national identity of the first creation, rather than attribute characteristic features that are not typical of it.

References

1. Alimov V.V. Translation theory. Translation in the field of professional communication: [Textbook] / V.V. Alimov. – M.: Nauka, 2005. – 225 p.
2. Ginzburg R.S. A Course in Modern English Lexicology / R.S. Ginzburg. – Moscow: Higher School Publishing House, 1966. – 209 p.
3. Granger S. Phraseology: an interdisciplinary perspective / S. Granger [edited by Sylviane Granger, Fanny Meunier]. – N.Y.: John Benjamins P.C., 2008. – 422 p.

Alina Karpenko

Vasyl' Stus Donetsk National University

Vinnytsia

Research Supervisor: V.I. Kalinichenko, PhD in Philology, Ass.Prof.

Language Advisor: Ia.V. Hryhoshkina, PhD in Philology, Ass. Prof.

WHY DO WE NEED WORD EMBEDDING, IT'S PROPERTIES AND FUTURE DIRECTIONS

Introduction. Natural language processing (hereinafter – NLP) with deep learning is an important combination in the modern world. Using word vector representations and embedding layers you can train recurrent neural networks with outstanding performances in the wide variety of industries. Word embeddings (hereinafter – WE), which encode meanings of words to low-dimensional vector spaces, have become very popular due to their state-of-the-art performances in many NLP tasks.

Word embeddings are substantially successful in capturing semantic relations among words, so a meaningful semantic structure must be present in the respective vector spaces. However, in many cases, this semantic structure is broadly and heterogeneously distributed across the embedding dimensions making interpretation of dimensions a real challenge.

The goal of word-embedding algorithms is, therefore, to embed words with the meaning based on their similarity or relationship with other words. There are a lot of applications made possible by word embeddings, also we can learn from the way researchers approached the problem of deciphering natural language for machines.

Review of recent publications. I have made a review of the most significant publications and methods. It is no exaggeration to say that word embeddings have revolutionized NLP. From early distributional semantic models (Turney and Pantel; Erk; Clark), to deep learning based word embeddings (Collobert and Weston; Mikolov; Pennington; Bojanowski) [1; 6; 7]. Word-level meaning representations have found applications in a wide variety of core NLP tasks, to the extent that they are now ubiquitous in the field [2].

Objectives of the paper is to provide a short overview and draw a conclusion about the current situation in this field of science: focus on the deficiencies of word embeddings and how recent approaches have tried to resolve them, recent developments, trends and future directions in WE.

Results of the research. In word-embedding models, each word in a given language is assigned to a high-dimensional vector such that the geometry of the vectors captures semantic relations between the words, e.g. vectors being closer together has been shown to correspond to more similar words.

These models are typically trained automatically on large corpora of texts, such as collections of Google News articles or Wikipedia, and are known to establish relations not found through simple co-occurrence analysis. For example, the vector for Ukraine is close to vectors for France and Spain, and the vector for XBox is close to that of PlayStation.

Beyond nearby neighbors, embeddings can also capture more global links between words. The difference between London and England, obtained by simply subtracting these two vectors, is parallel to the vector difference between Paris and France. This pattern allows embeddings to capture analogy relationships, such as London to England is as Paris to France [1].

Unlike word vectors obtained via one-hot encoding, word embeddings are learned from data. It is common to see word embeddings that are 256-dimensional, 512-dimensional, or 1024-dimensional when dealing with very large vocabularies. On the other hand, one-hot encoding words generally leads to vectors that are 20,000-dimensional or higher (capturing a vocabulary of 20,000 token in this case). So, word embeddings pack more information into far fewer dimensions.

There are two ways to obtain word embeddings:

1. Learn word embeddings jointly with the main task you care about (e.g. document classification or sentiment prediction). In this setup, you would start with

random word vectors, then learn your word vectors in the same way that you learn the weights of a neural network.

2. Load into your model word embeddings that were pre-computed using a different machine learning task than the one you are trying to solve. These are called 'pre-trained word embeddings'.

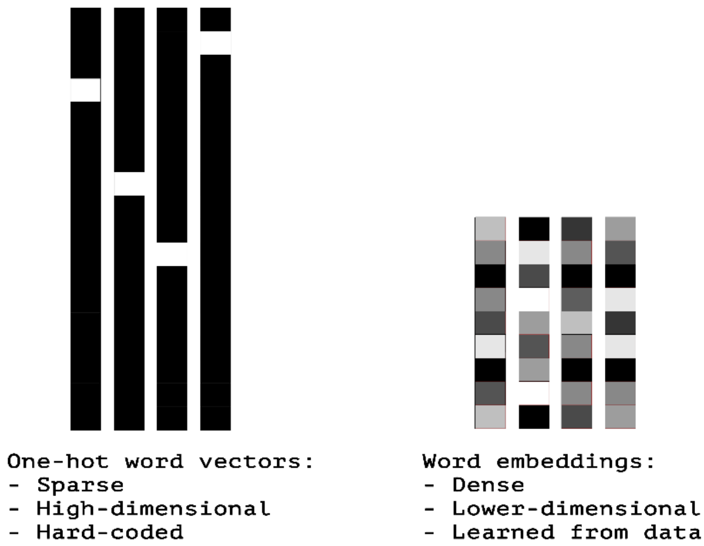

Fig. 1 one hot encoding vs word embeddings

The most popular word embedding model is highlighted below, the model that launched a thousand word embedding papers: word2vec, the subject of two papers by Mikolov and others in 2013. As word embeddings are a key building block of deep learning models for NLP, word2vec is often assumed to belong to the same group. Technically however, word2vec is not be considered to be part of deep learning, as its architecture is neither deep nor uses non-linearities (in contrast to Bengio's model and the C&W model) [2; 6]. Word2vec is a two-layer shallow neural net, so it is not an example of deep learning. Nevertheless, techniques like Word2vec and Global Vectors (hereinafter – GloVe) can turn a raw text into a numerical form that deep nets can understand.

Any individual programmer or scholar can use these tools and contribute new knowledge. Many areas of research and industry that could benefit from NLP have yet to be explored. Word embeddings and neural language models are powerful techniques. None the less, the most powerful aspect of machine learning may be its collaborative culture. Many, if not all, state-of-the-art methods are open-source, along with their accompanying research.

It is common for researchers to make pre-trained word embeddings available for free, often under a permissive license so that you can use them on your own academic or commercial projects. For example, both word2vec and GloVe word embeddings are available for free download. These can be used in your project instead of training your own embeddings from scratch. You have two main options when it comes to using pre-trained embeddings:

2. Static, where the embedding is kept static and is used as a component of your model. This is a suitable approach if the embedding is a good fit for your problem and gives good results.

3. Updated, where the pre-trained embedding is used to seed the model, but the embedding is updated jointly during the training of the model. This may be a good option if you are looking to get the most out of the model and embedding on your task [7].

Naturally, every feed-forward neural network that takes words from a vocabulary as input and embeds them as vectors into a lower dimensional space, which it then fine-tunes through back-propagation, necessarily yields word embeddings as the weights of the first layer, which is usually referred to as the Embedding Layer. The main difference between such a network that produces word embeddings as a by-product and a method, such as word2vec, with the generation of word embeddings as the explicit goal, includes its computational complexity. Generating word embeddings with a very deep architecture is simply too computationally expensive for a large vocabulary. This is the main reason why it has taken until 2013 for word embeddings to explode onto the NLP stage; computational complexity is a key trade-off for word embedding models, and it will be a recurring theme in the given review.

Another difference is the training objective: word2vec and GloVe are geared towards producing word embeddings that encode general semantic relationships, which are beneficial to many downstream tasks; notably, word embeddings trained this way won’t be helpful in tasks that do not rely on these kind of relationships. In contrast, regular neural networks typically produce task-specific embeddings that are only of limited use elsewhere. Note that a task that relies on semantically coherent representations, such as language modelling, will produce similar embeddings to word embedding models.

GLoVe is another method for deriving word vectors. It doesn’t have an implementation in the popular libraries we’re used to, but they should not be ignored. The algorithm is derived from algebraic methods (similar to matrix factorization), performs very well, and it converges faster than Word2Vec. As a side-note, word2vec and Glove might be said to be to NLP what VGGNet is to vision, i.e. a common weight initialisation that provides generally helpful features without the need for lengthy .

FastText was developed by the team of Tomas Mikolov who offered the word2vec framework in 2013, triggering the explosion of research on universal word embeddings. The main improvement of FastText over the original word2vec vectors is the inclusion of character n-grams, which allows computing word representations for words that did not appear in the training data (‘out-of-vocabulary’ words).

The Deep Contextualized Word Representations (ELMo) have recently improved the state of the art in word embeddings by a noticeable amount. They were developed by the Allen institute for AI and will be presented at NAACL 2018 in early June. In ELMo, each word is assigned to a representation which is a function of the entire corpus sentences to which they belong. The embeddings are computed from the internal states of a two-layers bidirectional Language Model (LM), hence the name “ELMo”: Embeddings from Language Models [5].

Word embeddings are typically learned only based on the window of surrounding context words. Levy & Goldberg have shown that dependency structures can be used as a context to capture more syntactic word relations [2]; Köhn finds that such dependency-based embeddings perform best for a particular multilingual evaluation method that clusters embeddings along with different syntactic features [3].

Melamud et al. observes that different context types work well for different downstream tasks and that simple concatenation of word embeddings learned with different context types can yield further performance gains [5]. Besides selecting context words differently, additional context may also be used in other ways: Tissier et al. incorporate co-occurrence information from dictionary definitions into the negative sampling process to move related works closer together and prevent them from being used as negative samples [10]. We can think of topical or relatedness information derived from other contexts such as article headlines or Wikipedia intro paragraphs that could similarly be used to make the representations more applicable to a particular downstream task.

Conclusion. As NLP models are being increasingly employed and evaluated in multiple languages; creating multilingual word embeddings is becoming a more important issue and has received increased interest over recent years. A promising direction is to develop methods that learn cross-lingual representations with as few parallel data as possible, so that they can be easily applied to learn representations even for low-resource languages [6]. Word embeddings can be considered an integral part of NLP models. As an unsupervised learning technique, it can be trained on any corpus without the need for any human interference. They provide a nice starting point for training any neural network, taking text as its input (you will have to convert that to indices first) since they capture similarity and relations similar to the examples above. It is pleasant to observe that as a community we are progressing from applying word embeddings to every possible problem to gaining a more principled, nuanced, and practical understanding of them.

References

1. Collobert R. Natural Language Processing (Almost) from Scratch. – 2011. – Retrieved from: http://dl.acm.org
2. Goldberg Y., Levy O. Word2vec Explained: Deriving Mikolov et al.’s Negative Sampling Word-embedding Method. – 2014. – Retrieved from: arXiv:1402.3722
3. Köhn A. What’s in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. – Lisbon, Portugal (17-21 September 2015), 2014. – P. 2067–2073.

4. Levy O., Goldberg Y. Dependency-Based Word Embeddings. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), 2014. – P. 302–308.
5. Melamud O., McClosky D. The Role of Context Types and Dimensionality in Learning Word Embeddings / O. Melamud, D. McClosky, S. Patwardhan, M. Bansal // Proceedings of NAACL-HLT 2016. – 2016. – P. 1030–1040. – Retrieved from: http://arxiv.org/abs/1601.00893
6. Mikolov T., Yih W. Linguistic Regularities in Continuous Space Word Representations / Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig. – Vol. 13. – 2013.
7. Pennington J., Socher R., Manning C. Glove: Global Vectors for Word Representation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). – Doha, Qatar. Association for Computational Linguistics, 2014. – P. 1532–1543.
8. Ruder S., Vulić I., Søgaard A. A Survey of Cross-lingual Word Embedding Models Sebastian. – 2017 – Retrieved from: http://arxiv.org/abs/1706.04902
9. The Current Best of Universal Word Embeddings and Sentence Embeddings [Remote employment]. – Retrieved from: https://medium.com/huggingface/universal-word-sentence-embeddings-ce48ddc8fc3a
10. Tissier J., Gravier C., Habrard A. Dict2Vec: Learning Word Embeddings using Lexical Dictionaries // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. – 2017. – Retrieved from: http://aclweb.org/anthology/D17-1024

Natalia Kybaliuk,
Inna Dolia
Mykhailo Kotsiubynskyi State Pedagogical University
Vinnytsia
Ressearch Supervisor: I.V. Dolia, PhD in Philology, Senior Lecturer
Language Advisor: I.V. Dolia, PhD in Philology, Senior Lecturer

DISCOURSE AS THE MAIN UNIT OF PRAGMATIC LINGUISTICS

Introduction. One of the most important issues in the field of modern pragmatic linguistics is the question of discourse. Nowadays it is difficult to give a clear definition to this concept taking into account the broad history of the discourse text formation and its ambiguous position in the system of existing categories. The ambiguity of the notion is determined by the history of its formation and, to a certain extent, the uncertainty of its place in the system of language realization existing categories.

The term ‘discourse’ comes from the Latin word ‘discursus’ which means ‘to wander’. Later on it acquires a few other meanings. ‘Discourse’ in the English