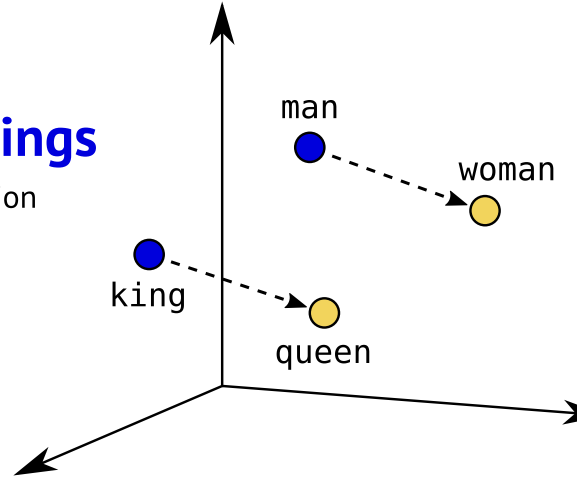
Enhancing word embeddings

Positionality, subword sizes, and hyphenation



Vítek Novotný
witiko@mail.muni.cz

Faculty of Informatics, Masaryk University

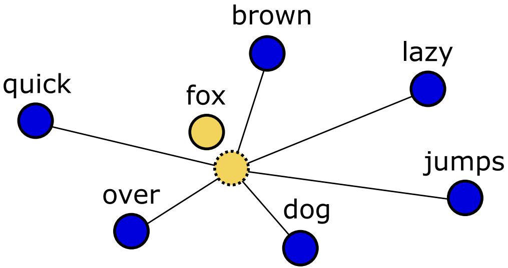
October 13, 2020

Introduction

- SOTA DNN LMs are accurate but also slow, black-boxed, and monolithic.
- Word embeddings of SNN LMs [1–4] provide strong baselines for many tasks:
  1. semantic text similarity [5]
  2. text classification [6]
  3. information retrieval [7]
- Word embeddings produce systems that are fast, interpretable, and modular.
- MIR@MU research group develops and maintains Gensim [8]:
  - Essential Python NLP library: 2.6k article citations and 11.2k stars on GitHub
  - Contains hardware-accelerated implementation of Word2Vec and fastText SLL LMs. [9]
  - Perfect tool to prototype, implement, and evaluate enhanced word embeddings.

Positionality

- Word2Vec [1, 2] and fastText [4] CBOW models are trained to minimize the distance between the mean of context word embeddings and the masked word embedding:

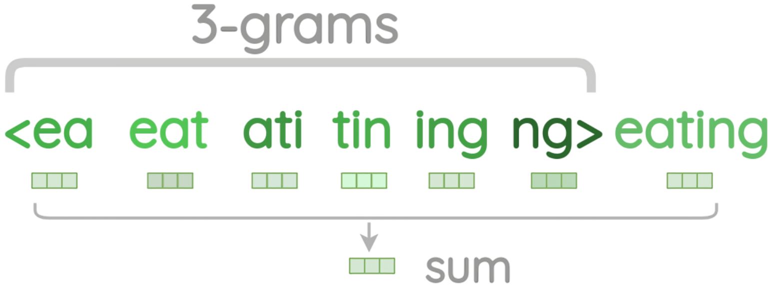


The quick brown **???** jumps over the lazy dog.

- However, the position of words in context is not taken into account.
- Mikolov et al. [10] achieved SOTA on English Word Analogy task using position-dependent weighting. However, no open-source implementation exists.

Subword sizes

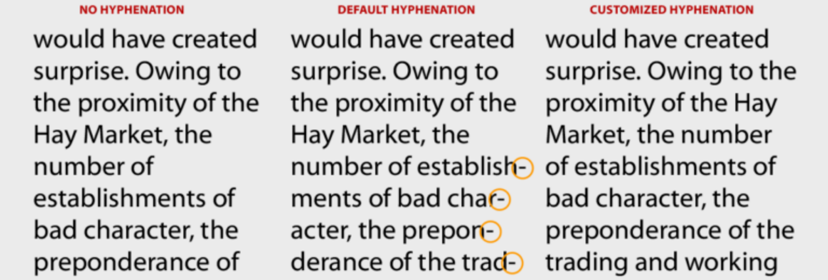
- Unlike Word2Vec [1, 2], fastText [4] embeds not only words, but also subwords.



- This speeds up training and allows inference of embeddings for unknown words.
- However, previous work reports optimal subword sizes only for English and German.
- Our experiments suggest:
  1. 5% improvement on Czech Word Analogy task with optimal subword sizes over defaults.
  2. A fast method for estimating the optimal subword sizes from corpus statistics.

Hyphenation

- Hyphenation splits words into subwords based on morphology or phonology.



NO HYPHENATION	DEFAULT HYPHENATION	CUSTOMIZED HYPHENATION
would have created	would have created	would have created
surprise. Owing to	surprise. Owing to	surprise. Owing to the
the proximity of the	the proximity of the	proximity of the Hay
Hay Market, the	Hay Market, the	Market, the number
number of	number of establish-	of establishments of
establishments of	ments of bad char-	bad character, the
bad character, the	acter, the prepon-	preponderance of the
preponderance of	derance of the trad-	trading and working

- TEX's hyphenation algorithm [11] achieves perfect accuracy with tiny models. [12, 13]
- fastText [4] embeds only subwords of fixed size and ignores morphology.
- Hyphenating fastText should decrease model size and speed up training.

Sounds fun?

- Take a look at our bachelor's and master thesis topics:
  - Positional weighting of fastText word embeddings (bachelor's thesis, diploma thesis)
  - Finding optimal n-gram sizes for fastText Models (bachelor's thesis, diploma thesis)
  - ... or come up with your own thesis topic!
- Join us at the PV212 seminar this Thurstday at 10 AM (CET) over Zoom, where we will dive into the details of out word embedding experiments.

Bibliography I

[1] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[2] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[3] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[4] Piotr Bojanowski et al. "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

Bibliography II

[5] Delphine Charlet and Geraldine Damnati. “Simbow at semeval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 2017, pp. 315–319.

[6] Matt Kusner et al. “From word embeddings to document distances”. In: *International conference on machine learning*. 2015, pp. 957–966.

[7] Vít Novotný et al. “Three is Better than One. Ensembling Math Information Retrieval Systems”. In: *CEUR Workshop Proceedings*. ISBN: 1613-0073.

[8] Radim Rehurek and Petr Sojka. “Software framework for topic modelling with large corpora”. In: *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer. 2010.

Bibliography III

[9] Radim Řehůřek. *Word2vec Tutorial*. URL: `https://rare-technologies.com/word2vec-tutorial/` (visited on 10/12/2020).

[10] Tomas Mikolov et al. “Advances in pre-training distributed word representations”. In: *arXiv preprint arXiv:1712.09405* (2017).

[11] Franklin Mark Liang. *Word Hy-phen-a-tion by Com-put-er*. Tech. rep. Calif. Univ. Stanford. Comput. Sci. Dept., 1983.

[12] Petr Sojka and Ondřej Sojka. “The unreasonable effectiveness of pattern generation”. In: *Zpravodaj CSTUG* (2019).

[13] Petr Sojka and Ondrej Sojka. “Towards Universal Hyphenation Patterns.”. In: *RASLAN*. 2019, pp. 63–68.

MUNI
FACULTY
OF INFORMATICS