Bioinformatics, 37(16), 2021, 2499–2501
doi: 10.1093/bioinformatics/btaa995
Advance Access Publication Date: 27 November 2020
Applications Note

Data and text mining

LexExp: a system for automatically expanding concept lexicons for noisy biomedical texts

Abeed Sarker *

Department of Biomedical Informatics, School of Medicine, Emory University, Atlanta, GA 30322, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on May 31, 2020; revised on October 4, 2020; editorial decision on November 13, 2020; accepted on November 17, 2020

Abstract

Summary: LexExp is an open-source, data-centric lexicon expansion system that generates spelling variants of lexical expressions in a lexicon using a phrase embedding model, lexical similarity-based natural language processing methods and a set of tunable threshold decay functions. The system is customizable, can be optimized for recall or precision and can generate variants for multi-word expressions.

Availability and implementation: Code available at: https://bitbucket.org/asarker/lexexp; data and resources available at: https://sarkerlab.org/lexexp.

Contact: abeed@dbmi.emory.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Lexicon- or dictionary-based biomedical concept detection approaches require the manual curation of relevant lexical expressions, generally by domain experts (Demner-Fushman and Elhadad, 2016; Ghiassi and Lee, 2018; Rebholz-Schuhmann *et al.*, 2013; Shivade *et al.*, 2014). To aid the tedious process of lexicon creation, automated lexicon expansion methods have received considerable research attention, leading to resources such as the UMLS SPECIALIST system (McCray *et al.*, 1993), which is utilized in MetaMap (Aronson and Lang, 2010) and cTAKES (Savova *et al.*, 2010). Such systems perform well for formal biomedical texts, but not noisy texts from sources such as social media and electronic health records. Due to the presence of non-standard expressions, misspellings and abbreviations, it is not possible to capture all possible concept variants in noisy texts using manual or traditional lexicon expansion approaches. The number of lexical variants of a given concept that may occur, although finite, cannot be predetermined. Biomedical concepts, such as symptoms and medications, are specifically likely to be misspelled compared to non-biomedical concepts (Soualmia *et al.*, 2012; Zhou *et al.*, 2015). Despite advances in machine learning based sequence labeling approaches, which typically outperform lexicon-based approaches and can detect inexact concept expressions, the latter are frequently used in biomedical research. This is non-exclusively because machine learning methods require manually annotated datasets, which may be time-consuming and expensive to create, and training and executing state-of-the-art machine learning approaches may require technical expertise and high-performance computers, which may not be available. In this article, we describe an unsupervised lexicon expansion system (*LexExp*), which automatically generates many lexical variants of expressions encoded in a lexicon.

2 Materials and Methods

LexExp builds on recent studies, including our own, which utilize the semantic similarities captured by word2vec-type dense-vector models (Mikolov *et al.*, 2013) to automatically identify similar terms and variants (Percha *et al.*, 2018; Sarker and Gonzalez-Hernandez, 2018; Viani *et al.*, 2019). LexExp employs customizable *threshold decay* functions (*constant*, *linear*, *cosine*, *exponential*), combined with dense-vector and lexical similarities, to generate many variants of lexicon entries. Similarity thresholding for determining lexical matches is popular (Fischer, 1982); most approaches apply static thresholding while some recent studies have attempted to employ dynamic thresholding for misspelling correction or generation (Sarker and Gonzalez-Hernandez, 2018; Savary, 2002). However, there is no existing tool that enables the use of customizable thresholding options for these tasks. The objective of LexExp is to generate lexical variants of multi-word expressions using customized thresholding, not lexically dissimilar semantic variants (e.g. synonyms).

Given a lexicon entry, LexExp first generates word n-grams ($n = 1$ and 2) from the entry (for one-word expressions, only unigrams are generated). For each n-gram within the entry, a dense embedding model is used to retrieve n most semantically similar words/phrases using cosine similarity, if the n-gram is present in the model. Next, all the words/phrases whose semantic similarities with the n-grams are higher than a threshold are included as candidate variants. For each candidate, its Levenshtein ratio is computed against the original n-gram and a separate threshold for lexical similarity (t) is applied. All candidates below the threshold are removed from the list of possible variants. The same process is applied recursively on each remaining candidate until no new variants with

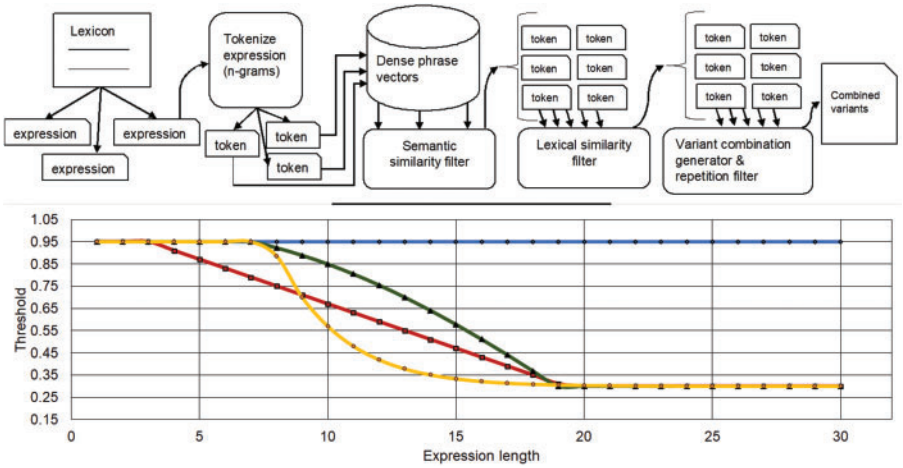

Fig. 1. (**top**) Workflow of the LexExp system. Expressions from the lexicon are first tokenized into *n*-grams. The dense-vector model is then used to generate semantically similar expressions, which are filtered to keep only *k* most semantically similar expressions for each source token. These expressions are then passed to the lexical similarity filter, which employs one of the threshold decay functions described in the article to remove too lexically dissimilar expressions. Finally, the generated variants are combined to produce all combinations of multi-word expressions and repetitions are removed. Such repetitions occur often and typically when a variant of an *n*-gram token has a different *n* in the variant. For example, "panic attack" is often expressed as one word, "panicattack", on social media (perhaps as a hashtag), which is generated as a semantic variant during the expansion process. The variant combination generator then combines the single-word expression with tokens in multi-word expressions leading to the generation of variants such as "panicattack attack". The repetition filter then attempts to identify such repetitions and remove them. (**bottom**) The four threshold determination functions that can be used in LexExp. Static—no change in threshold; linear—threshold decreases in a linear fashion with length; cosine—gradient of threshold decay increases with length and exponential—gradient of threshold decay decreases with length. For all these curves, $m = 2$, $n = 3$, initial threshold $(t_i) = 0.95$ and threshold lower bound $(t_l) = 0.30$

similarity above t are found. While we used an embedding model described in our past work (Sarker and Gonzalez, 2017), any can be used for identifying semantically similar terms in LexExp.

2.1 Lexical similarity thresholding functions

LexExp provides the user with four functions that can be used to vary t based on the character lengths of the input *n*-grams. Typically, longer terms/phrases have *true* variants that are lexically more distant from the original entry. So, adjusting t based on the length of an expression may lead to better precision and/or recall. Note that recall and precision are both ill-defined in this context: recall—because there is no known bound for the total number of variants; precision—because the set of *true variants* depends on the research task. Given an expression, p, the four functions are:

Static: $t = t_i$

Linear decay:

$$t = \max\left(t_i - \frac{m \times (\text{len}(p) - n)}{100.0}, t_l\right)$$

Cosine decay:

$$t = \min\left(\max\left(\cos\left(\frac{\max(\text{len}(p) - n, 0)}{m \times 2\pi}\right), t_l\right), t_i\right)$$

Exponential decay:

$$t = \min\left(\left(\text{len}(p) \times 2 \times m \times e^{(.5 \times \text{len}(p))} + \left(\frac{n}{10}\right)\right), t_i\right)$$

where m and n are constants, t_i is the initial threshold and t_l is the lower bound for t. Figure 1 illustrates how these thresholding methods vary for expressions of length 1–30 characters. These thresholding functions are carefully designed to provide the user with flexibility to vary them as per the needs of a task.

2.2 Multi-word variants

A key functionality of LexExp is its ability to generate variants for multi-word expressions. Capturing variants of multi-word expressions comprehensively is particularly challenging via manual annotation since the number of possible word combinations can be very high. Also, phrase embedding models cannot capture the semantics of long multi-word expressions due to the sparsity of their occurrences.

LexExp uses two functions for generating multi-word variants. The first is a unigram variant generation function that generates variants for each word based on a specific value of t, and then generates all combinations of the original expression based on the variants identified, keeping the ordering of the variants unchanged. Examples of variants generated by this function are shown below:

Original expression: eyes were excruciatingly sensitive and sore.

Sample variants:

1. eyes were **excrusiatingly sensistive** and sore;
2. eyes were excruciatingly **sensitve** and sore;
3. eyes were **excrusiatingly** sensitive and sore;
4. eyes were **excrusiatingly sensetive** and sore.

The second is a bigram generation function, which first tokenizes the expressions into bigrams, then generates variants of the bigrams. These variants maybe uni- or multi-grams (e.g. stomach ache: stomachache, mild stomach ache). After the variants are generated, they are tokenized to unigrams and then all combinations of all unigrams are generated as described before. Recombining the bigrams following the generation of the variants can be complicated in some cases, as a term and its *partial variant* may both be present in a combination (see Figure 1 caption). LexExp attempts to resolve these using a simple forward and backward pass through the list of words, removing all words identical to or substrings of the next/previous one.

3 Conclusion

We ran LexExp on multiple lexicons, including lexicons for COVID-19 symptoms from Twitter (Sarker *et al.*, 2020), adverse drug reactions (Sarker and Gonzalez, 2015), a subset of the consumer health vocabulary (Zeng and Tse, 2006) and psychosis symptoms from electronic health records (Viani *et al.*, 2019). We also compared tweet retrieval numbers for COVID-19 symptom-mentioning tweets using the abovementioned lexicon with and without variants, and observed an increase of 16.6%. Further details about these experiments are provided in Supplementary Material. As a lexicon expansion system, the purpose of LexExp is not to obtain perfect accuracy—in fact, accuracy is not well-defined for this

generation task. The objective, instead, is to automatically generate large sets of possible variants that can be readily used by human experts for information retrieval and extraction.

Funding

Research reported in this publication was supported by the National Institute on Drug Abuse (NIDA) of the National Institutes of Health (NIH) under award No. R01DA046619. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

Conflict of Interest

None declared.

References

Aronson,A.R. and Lang,F.-M. (2010) An overview of MetaMap: historical perspective and recent advances. *J. Am. Med. Inform. Assoc.*, **17**, 229–236.

Demner-Fushman,D. and Elhadad,N. (2016) Aspiring to unintended consequences of natural language processing: a review of recent developments in clinical and consumer-generated text processing. *IMIA Yearbook*, **25**, 224–233.

Fischer,R.-J. (1982) *A Threshold Method of Approximate String Matching*. Springer, Berlin, Heidelberg, pp. 843–849.

Ghiassi,M. and Lee,S. (2018) A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. *Expert Syst. Appl.*, **106**, 197–216.

McCray,A.T. *et al.* (1993) UMLS® knowledge for biomedical language processing. *Bull. Med. Libr. Assoc.*, **81**, 184–194.

Mikolov,T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems* - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, pp. 3111–3119.

Percha,B. *et al.* (2018) Expanding a radiology lexicon using contextual patterns in radiology reports. *J. Am. Med. Inform. Assoc.*, **25**, 679–685.

Rebholz-Schuhmann,D. *et al.* (2013) Evaluating gold standard corpora against gene/protein tagging solutions and lexical resources. *J. Biomed. Sem.*, **4**, 28.

Sarker,A. *et al.* (2020) Self-reported COVID-19 symptoms on Twitter: an analysis and a research resource. *J. Am. Med. Inform. Assoc.*, **27**, 1310–1315.

Sarker,A. and Gonzalez,G. (2015) Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *J. Biomed. Inform.*, **53**, 196–207.

Sarker,A. and Gonzalez,G. (2017) A corpus for mining drug-related knowledge from Twitter chatter: language models and their utilities. *Data Brief.*, **10**, 122–131.

Sarker,A. and Gonzalez-Hernandez,G. (2018) An unsupervised and customizable misspelling generator for mining noisy health-related text sources. *J. Biomed. Inform.*, **88**, 98–107.

Savary,A. (2002) *Typographical Nearest-Neighbor Search in a Finite-State Lexicon and its Application to Spelling Correction. Lecture Notes in Computer Science. Artificial Intelligence and Lecture Notes in Bioinformatics*. Springer Verlag, Berlin, Heidelberg, pp. 251–260.

Savova,G.K. *et al.* (2010) Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J. Am. Med. Inform. Assoc.*, **17**, 507–513.

Shivade,C. *et al.* (2014) A review of approaches to identifying patient phenotype cohorts using electronic health records. *J. Am. Med. Inform. Assoc.*, **21**, 221–230.

Soualmia,L.F. *et al.* (2012) Matching health information seekers' queries to medical terms. *BMC Bioinform.*, **13**, S11.

Viani,N. *et al.* (2019) *Generating Positive Psychosis Symptom Keywords from Electronic Health Records. Lecture Notes in Computer Science. Artificial Intelligence and Lecture Notes in Bioinformatics*. Springer Verlag, Berlin, Heidelberg, pp. 298–303.

Zeng,Q.T. and Tse,T. (2006) Exploring and developing consumer health vocabularies. *J. Am. Med. Inform. Assoc.*, **13**, 24–29.

Zhou,X. *et al.* (2015) Context-sensitive spelling correction of consumer-generated content on health care. *JMIR Med. Inform.*, **3**, e27.