LEARNING DOMAIN SPECIFIC SEMANTIC WORD RELATIONSHIPS FROM MEDICAL DOCUMENTS

DAVID CALLENDER, RUI CHEN, ZHENSHUAI DING

1. Introduction & Motivation

We propose using unsupervised machine learning algorithms, specifically word2vec [2] and Latent Dirichlet Allocation (LDA) [1], to learn continuous abstract feature vectors (e.g. $w \in \mathbb{R}^n$) representing words in a vocabulary by analyzing text corpora specific to the bio-medical field. It has been shown that the vector representations produced by these algorithms can be used to predict word analogies consisting of pairs of semantically related words [4] by leveraging consistent linear relationships between pairs of words linked by a specific concept. In particular, such pairs of words exhibit similar vector offsets (as measured by cosine distance between the offsets).

Prior work [2] has been focused on very general text documents such as assortments of news articles on wide ranging topics. Therefore, the semantic relationships tested thus far have also been general. Examples include countries/states and the respective capital cities, currencies, or gender information. We propose testing these algorithms ability to learn concepts from a specific domain. One such domain for which much work has been done on cataloging semantic links is the medical field. Vast ontologies have been hand engineered into semantic networks of medical terms. We propose extracting examples from the Unified Medical Language System (UMLS) of semantically related words from a selection of classes of semantic concepts that we will use to evaluate the effectiveness of our unsupervised learning algorithms.[1]

2. Methods & Techniques

2.1. **Training Algorithm.**

We propose focusing on two unsupervised algorithms: word2vec and LDA. These are two very different algorithms and unfortunately differ greatly in their running time efficiency. However, they both can be used to produce abstract feature vectors for words in a predetermined vocabulary.

2.1.1. *word2vec.* The former algorithm is a simple neural network consisting of a projection layer that takes as input one-hot representations of vocabulary words and outputs vectors $w_i \in \mathbb{R}^n$, where i is the index of the corresponding vocabulary word that the w_i represents. During training the projection layer output is used as input to a classification layer similar to softmax but much less computationally expensive. Both hierarchical softmax and negative sampling have been shown to function well as a classifier in word2vec. The classification layer is discarded after training and the vectors produced by the projection layer are retained.

Date: October 2, 2014.

[1] Unified Medical Language System (UMLS): http://www.nlm.nih.gov/research/umls/

There are two implementations of word2vec that we can adapt to our project. The first was developed by the authors who proposed the algorithm.[2] It is written in C++ and has been shown to be capable of analyzing text corpora on the order of billions of words in about an hour when run on a single multicore computer. The second implementation is a port of the C++ code to python.[3] Because it has optimized using Cython it has been shown to be similarly efficient. Our choice in implementations will depend largely on to what extent we need to modify the algorithms for our proposed problem. Likely, we will use the python implementation.

2.1.2. *Latent Dirichlet Allocation (LDA).* LDA is a popular topic model [5] that can reveal hidden topics from text documents using unsupervised learning. LDA assumes that each word in the vocabulary has a latent topic drawn from a probability distribution of topics. This specific set of probabilities of topics for a word, once learned, can be used as a feature vector for that word that incorporates semantic information about that word in the form of affinity to abstract topics. Topic labels are not specified, rather a choice of the number of latent topics is made. This number corresponds to the dimension of the abstract feature vectors.

Unfortunately, while there are many efficient implementations of LDA available, the complexity of the algorithm itself means that it inherently can not handle corpora of the same size that word2vec can. Therefore, our evaluation of LDA will be limited to fewer words then we can achieve with word2vec. We hope to be able to compare LDA to word2vec on smaller corpora and then extrapolate to larger data sets based on the word2vec performance.

2.2. **Evaluation metric: accuracy.** To evaluate the ability of these algorithms to capture binary semantic relationships we will utilize the accuracy metric proposed by Mikolov et al. [4, 2]. The accuracy is measured by randomly drawing two (a and b) pairs of words from the same semantic relationship class S. If a consists of the ordered pair of words, (w_{a1}, w_{a2}), and b consists of the pair (w_{b1}, w_{b2}); then, we can quantify wether their respective vector offsets are similar (e.g. $w_{a1} - w_{a2} \approx w_{b1} - w_{b2}$) via an all or nothing accuracy function, $f(a, b)$ defined in Equation (1).

$$f(a,b) = \begin{cases} 1 & \text{if } w_{b1} = \text{argmin}_{w \in \text{vocabulary}} \, ||(w_{a1} - w_{a2}) - (w - w_{b2})|| \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The overall accuracy is then simply the average all or nothing accuracy over repeated drawings of one or more semantic relationship classes. Using this measure allows for comparison to previous results[2].

3. Training Data

3.1. **Text Corpus.** Our primary source of text documents will be abstracts pulled from PubMed. We have chosen this resource as it is possible to download large quantities of targeted references (including abstracts) from PubMed.[4] Furthermore, abstracts are free of unwanted components such as charts and figures. Searching for the term ‘medicine’ on PubMed, while limiting the results to entries containing abstracts, yields 2,680,182 results. If the abstracts are on average at least 375 words

[2] https://code.google.com/p/word2vec/

[3] http://radimrehurek.com/2013/09/deep-learning-with-word2vec-and-gensim/

[4] http://www.ncbi.nlm.nih.gov/pubmed

this would yield a data set in excess of 1 billion words, which is on par with that used by Mikolov et al.[2]. The results of such a search can be downloaded in XML format. It will then be necessary to pull out the abstracts and most likely clean the text.

3.2. **Known binary semantic concepts.** The UMLS (Unified Medical Language System) contains a rich network of medical terms connected by labeled binary (two words, one edge) semantic relationships. The primary relationship is a parent to child connection labeled 'is a', thus creating a tree like structure. However, there additionally are a rich set of more interesting relations encapsulating detailed concepts pertaining to physical, spatial, functional, temporal, and other medically relevant conceptual relationships.[5] In fact, these relationships have there own hierarchy.

A particular challenge of this project will be choosing which semantic concepts to evaluate and determining good candidates of word pairs for those concepts. Many terms in the UMLS consist of more than one word. Likely, we will limit our choices to edges where the connected terms are both single words.

Additionally, extracting this information from the UMLS will be challenging. It is a complicated data structure. The UMLS Terminology Services (UTS), provides access to the UMLS semantic network.[6] Access is provided in three forms that will be of interest to us. First, a web app browser can be used for exploration. Second, an API is provided for automated web queries. Lastly, the data files can be downloaded that contain the UMLS in its entirety. These last two forms of access is what we will utilize to create our target analogies for evaluating the accuracy of the unsupervised machine learning algorithms.

4. Timeline & Milestone Goals

Data selection is a critical step of any research and scientific investigation. While we plan to use the PubMed source of abstracts initially, we will also be evaluating other data sources, possibly other sources of journal articles and conference papers. Additionally, we have an XML file obtained from the US government's online repository of Clinical trials.[7] This file has approximately 1 GB of text data consisting of descriptions of Clinical Trials, past and present.

Choosing a suitable size of our dataset is also a vital part of our task. If our dataset is too large, it will cost us too much time, and if the dataset is not large enough, we will not be able to train our models well. We will attempt to create data sets that have on the order of billions of words. This way we can truncate for speed and testing. Additionally, we have to do a wide variety of work on data processing. For example, in order to obtain LDA training data, we need to remove the stop-words from the text, do stemming on the text and convert the text into "bag-of-word" format.[6] Therefore, our primary milestone goal is to train our model on a decent, appropriate and satisfying dataset for our problem. Specifically, we will:

(1) Create sets of word pairs with known semantic connection by extracting them from UMLS.

[5] www.nlm.nih.gov/research/umls/META3_current_relations.html

[6] https://uts.nlm.nih.gov/home.html

[7] https://clinicaltrials.gov

(2) Collect raw text data, the abstracts from medical journals or conference paper. (shoot for 1 billion words).
(3) Write tools to extract abstracts from text and format the data for learning software.
(4) Based on word frequencies in abstracts and on word pairs from UMLS, create our master vocabulary (list of words we care about).
(5) Get word2vec implementation functioning on our corpus of medical paper abstracts.

Furthermore, we have our secondary goals

(1) Convert text data into bag-of-words format so that LDA model can also work on our dataset.
(2) Write tools using LDA model to work on the same function as word2vec model.
(3) Initial test for accuracy starting with smaller vocabulary and small corpus.
(4) Compare word2vec and LDA results on various size corpora and vocabulary to each other and to results from word2vec paper.

To sum up, at the time of milestone, we will setup our training data, figure out the useful vocabulary, implement tools to run basic tests on our data and improve their accuracy.

References

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 2013.

[4] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL HLT*, 2013.

[5] Christos Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis (postscript). 1998.

[6] Josef Sivic. Efcient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:591–605, 2009.