Efficient semantic indexing via neural networks with dynamic supervised feedback

Vivek Dhand

Commwealth Computer Research, Inc.
vivek.dhand@ccri.com

Abstract

We describe a portable system for efficient semantic indexing of documents via neural networks with dynamic supervised feedback. We initially represent each document as a modified TF-IDF sparse vector and then apply a learned mapping to a compact embedding space. This mapping is produced by a shallow neural network which learns a latent representation for the textual graph linking words to nearby contexts. The resulting document embeddings provide significantly better semantic representation, partly because they incorporate information about synonyms. Query topics are uniformly represented in the same manner as documents. For each query, we dynamically train an additional hidden layer which modifies the embedding space in response to relevance judgements. The system was tested using the documents and topics provided in the Total Recall track.

1 Introduction

We present a dynamic neural-network based system for portable semantic indexing of text documents to aid in technology-assisted review. Our starting point is the TF-IDF statistic, which is widely used in information retrieval to score the words in a document in terms of relevance and distinctiveness. These scores are then used to represent each document as a sparse vector. By interpreting TF-IDF in terms of graph theory, we are led to incorporate a global statistic for ranking the importance of words, and we modify the sparse document vectors accordingly. We then apply a neural network learning algorithm to represent words as dense vectors in a relatively low-dimensional semantic embedding space. As a result, any block of text can be represented as a sparse vector and then passed through the projection mapping to embedding space. Note that we do not make use of any external language resources or domain-specific knowledge

during this process.

Given an query topic, we use semantic search within embedding space to construct a seed set of documents for review. Supervised feedback in the form of relevance judgements is used to train an additional lightweight neural network. Any subsequent searches are performed inside the expanded embedding space corresponding to the hidden layer of the network.

2 Graph theoretic statistics: TF-IDF and LF-IDF

Let $G = (V, E)$ be a bipartite graph with vertex set $V = X \sqcup Y$ and edge set $E \subset X \times Y$. Let $f : E \to \mathbb{R}_{>0}$ be a function assigning positive real weights to the edges of G. We define several statistics associated to the pair (G, f). Note that these functions are asymmetric in X and Y, so we only present one-sided definitions for simplicity. Also, for ease of notation we write $x \sim y$ when $(x, y) \in E$.

Given $x \in X$, the *inverse document frequency* of x is defined as:

$$\mathrm{IDF}(x) = \log\left(\frac{|Y|}{\deg(x)}\right)$$

and the *global frequency* of x is defined as:

$$\mathrm{GF}(x) = \sum_{x \sim y} f(x, y).$$

Given $y \in Y$, the *maximum weight* of y is defined to be:

$$\mathrm{M}(y) = \max_{x \sim y} f(x, y)$$

Given an edge $(x, y) \in E$, we define the *term frequency* of x relative to y to be:

$$\mathrm{TF}(x, y) = f(x, y)/\mathrm{M}(y)$$

The well-known *term frequency - inverse document frequency* statistic is defined as:

$$\mathrm{TFIDF}(x, y) = \mathrm{TF}(x, y) \cdot \mathrm{IDF}(x).$$

The TFIDF statistic can be thought of as providing new weights on the graph G which better express the importance of various edges. Since TFIDF depends on the local statistic TF, it is natural to define a global version which involves summing over the weights of edges incident on a vertex x. In this way, we obtain a global version of TFIDF, which we call *log frequency - inverse document frequency*:

$$\mathrm{LFIDF}(x) = \log(1 + \mathrm{GF}(x)) \cdot \mathrm{IDF}(x).$$

In this paper, we use the product of TFIDF and LFIDF to rank edges.

2.1 Example: words and documents.

Let Y be a document corpus and let X denote the set of words contained in the corpus. If a word $x \in X$ is contained in a document $y \in Y$, we add the edge (x, y) and set $f(x, y)$ to be the frequency of x in y. In this case, the TFIDF function corresponds to the standard TF-IDF statistic, which yields a surprisingly good baseline for semantic indexing of text documents. However, it is possible for a globally rare word to have an artificially high TF-IDF value in a given document relative to the theme of the document. Working on the assumption that the the thematically important words in a document will be shared across documents, we augment the TFIDF function by multiplying it by LFIDF. This modification partially remedies this problem by giving a boost to words that appear in a relatively small number of documents but with relatively high global frequency.

2.2 Example: bigrams.

Let $X = Y$ be the set of words contained in a document corpus. For any bigram (x, y) which appears in the corpus text m times, we add an edge (x, y) with weight m. In this case, the value $TFIDF(x, y) \cdot LFIDF(x)$ ranks the words y by their affinity for appearing immediately after x in the text. We can also reverse the roles of X and Y, and thereby rank the words x by their affinity for appearing immediately before y in the text. By multiplying these values together, we obtain a symmetric function which expresses the internal affinity for each bigram (x, y). The resulting scores can be used to automatically annotate multi-word idiomatic phrases or proper names. Model improvements resulting from these annotations will be assessed in future work.

3 Semantic representation of words and documents

Let X and Y denote the vertices in the word-document graph from Example 2.1. Each document $y \in Y$ can be represented as a sparse vector $\mathbf{s}(y) \in \mathbb{R}^{|X|}$ whose value at a word x is equal to $TFIDF(x, y) \cdot LFIDF(x)$, if x occurs in y, and zero otherwise. Given $y_1, y_2 \in Y$, the cosine similarity of $\mathbf{s}(y_1)$ and $\mathbf{s}(y_2)$:

$$\text{sim}(y_1, y_2) = \frac{\mathbf{s}(y_1) \cdot \mathbf{s}(y_2)}{|\mathbf{s}(y_1)||\mathbf{s}(y_1)|}$$

gives a rough measure of the semantic similarity between the documents. However, this representation is clearly lacking: two semantically related documents could simply use

different words or phrases and their sparse vectors would have low cosine similarity. A standard approach is to apply dimensionality reduction algorithms to map from $\mathbb{R}^{|X|}$ to a more manageable embedding space $\mathbb{R}^d$ and hope that the compression captures latent semantic information. Rather than working with the set of document vectors directly, we propose to learn the projection by associating a semantic vector $\mathbf{v}(x)$ to each word $x \in X$. To this end, we make use of the skip-gram word embedding model contained in *word2vec* [2]. This algorithm efficiently produces clusters of synonyms in $\mathbb{R}^d$ and organizes them by type to some extent. The embedding vector of a document y is then defined as:

$$\mathbf{v}(y) = \sum_{x \in y} TFIDF(x, y)\, LFIDF(x)\, \mathbf{v}(x).$$

In our experiments, we apply a minimal amount of data cleaning to the input text. We lowercase the text, remove all non-alphanumeric characters, and then replace each digit with the # symbol. With enough training, the embedding vectors $\mathbf{v}(y)$ outperform the sparse vectors $\mathbf{s}(y)$ in terms of precision and recall. For example, below we plot the interpolated precision and recall curves comparing the two algorithms as measured on a sample from the *oldreut* Reuters corpus (Fig. 1).

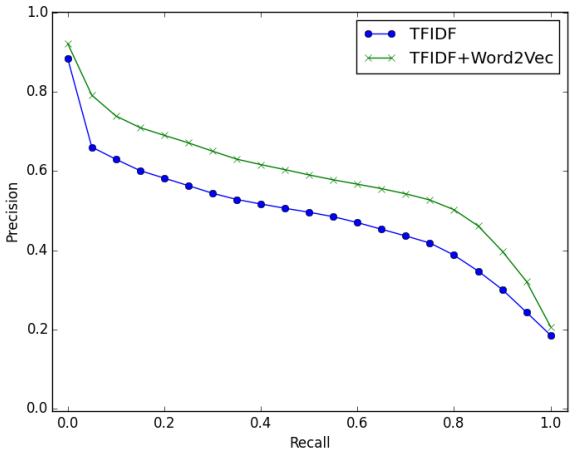

Figure 1: precision vs. recall, oldreut

4 Dynamic supervised feedback

Given a query topic, we pass the text of the query through the above process to produce a topic embedding vector $\mathbf{t}$. We then sort the document embeddings $\mathbf{v}(y)$ by cosine

similarity to $\mathbf{t}$, and return the k-nearest neighbors. Given a relevance judgement for each of these documents relative to the topic, we train a lightweight neural network with $2d$ neurons in the hidden layer and one output neuron which predicts the probability that a given embedding vector is relevant to the topic. Note that the number of parameters in this neural network is $O(d^2)$. In practice, good semantic representation can be achieved for relatively small values of d, so these classifiers are quite efficient in terms of space and training time. To generate the next document recommendations, we find the k-nearest documents to the image of $\mathbf{t}$ in the expanded embedding space $\mathbb{R}^{2d}$ and remove any documents that have already been viewed. Any further relevance judgements provide more training data for the neural network, which refines the hidden layer semantic search.

We ran the above system on three corpora, *athome1*, *athome2*, and *athome3*, with 10 query topics each. The number of documents in each corpus were approximately 290k, 460k, and 900k, respectively. The experiments were performed on a single node with 8 CPU cores and 16GB of RAM. For each corpus, 50-dimensional word embeddings were trained for 10 epochs, where an epoch is defined as one read through the files. For each topic, a classifier neural network with a 100-dimensional hidden layer was trained for 10 epochs, each time sampling up to 5,000 random training points. The size of the batches submitted for assessment was set to the nearest power of 10 less than or equal to the number of documents reviewed, up to a maximum batch size of 2,000 documents.

The recall as a function of review effort is plotted below, organized by corpus (Fig. 2, Fig. 3, Fig. 4), along with the text corresponding to each topic code.

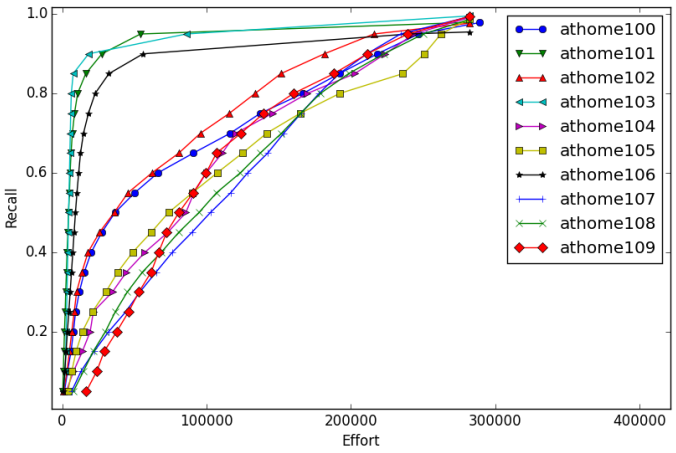

Figure 2: recall vs. effort, athome1

athome100	School and Preschool Funding
athome101	Judicial Selection
athome102	Capital Punishment
athome103	Manatee Protection
athome104	New medical schools
athome105	Affirmative Action
athome106	Terri Schiavo
athome107	Tort Reform
athome108	Manatee County
athome109	Scarlet Letter Law

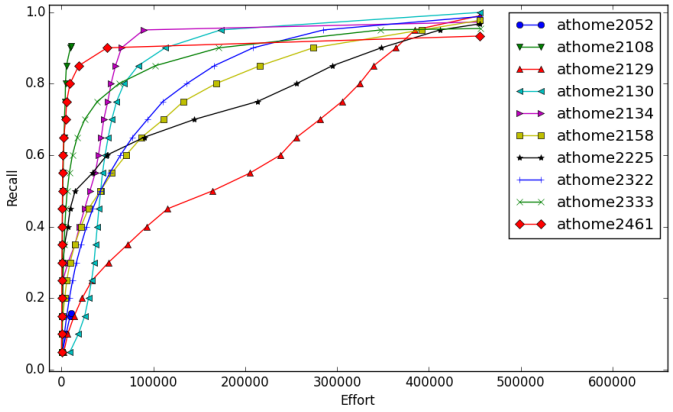

Figure 3: recall vs. effort, athome2

athome2052	Paying for Amazon book Reviews
athome2108	CAPTCHA Services
athome2129	Facebook Accounts
athome2130	Surely Bitcoins can be Used
athome2134	PayPal Accounts
athome2158	Using TOR for Anonymous Internet Browsing
athome2225	Rootkits
athome2322	Web Scraping
athome2333	Article Spinner Spinning
athome2461	Offshore Host Sites

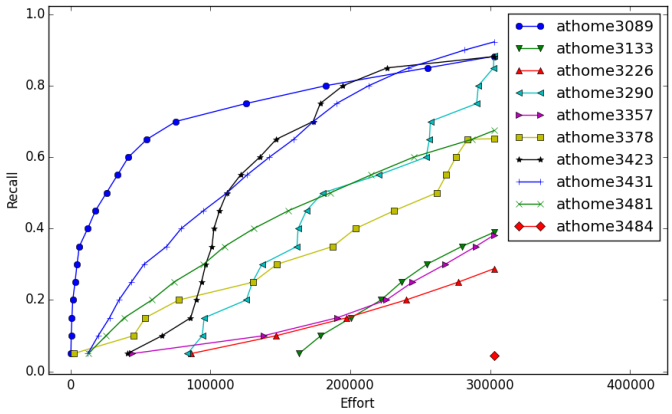

Figure 4: recall vs. effort, athome3

athome3089	Pickton Murders
athome3133	Pacific Gateway
athome3226	Traffic Enforcement Cameras
athome3290	Rooster Turkey Chicken Nuisance
athome3357	Occupy Vancouver
athome3378	Rob McKenna Gubernatorial Candidate
athome3423	Rob Ford Cut the Waist
athome3431	Kingston Mills Lock Murders
athome3481	Fracking
athome3484	Paul and Cathy Lee Martin

5 Discussion

While our seed model achieves superior semantic representation relative to TF-IDF, the dynamic component has some issues that inhibit performance, as compared to the Baseline Model Implementation (BMI) for continuous active learning (CAL) described in [1]. If the text of the query topic does not contain sufficiently distinctive words, then the topic embedding will not adequately capture the composite meaning of the topic text, to the detriment of the seed recommendations. Annotation of named entities and idiomatic phrases, e.g. as described in Example 2.2, would partly alleviate this problem.

Additional problems arise when only a miniscule number of documents in the corpus are relevant to a topic, since training a classifier requires positive examples. In this case, it becomes necessary to validate the seed recommendations by incorporating complementary methods, e.g. keyword search, before submitting them for assessment. It would also be helpful to adjust the classifier algorithm so that the training data is not overwhelmed by negative examples.

6 Acknowledgments

The author would like to thank T. Emerick and K. Sadeghi for sharing their insights during many helpful discussions. Thanks are also due to the CCRi leadership for supporting this research effort.

References

[1] G. V. Cormack and M. R. Grossman. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 153162. ACM, 2014.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR Workshop*, 2013.