Sequence-to-Set Semantic Tagging for Complex Query Reformulation and Automated Text Categorization in Biomedical IR using Self-Attention

Manirupa Das[†], **Juanxi Li**[†], **Eric Fosler-Lussier**[†],
Simon Lin[‡], **Steve Rust**[‡], **Yungui Huang**[‡] & **Rajiv Ramnath**[†]
The Ohio State University[†] & Nationwide Children's Hospital[‡]
{das.65, li.8767, fosler.1, ramnath.6}@osu.edu
Simon.Lin, Steve.Rust, Yungui.Huang}@nationwidechildrens.org

Abstract

Novel contexts, comprising a set of terms referring to one or more concepts, may often arise in complex querying scenarios such as in evidence-based medicine (EBM) involving biomedical literature. These may not explicitly refer to entities or canonical concept forms occurring in a fact-based knowledge source, e.g. the UMLS ontology. Moreover, hidden associations between related concepts meaningful in the current context, may not exist within a single document, but across documents in the collection. Predicting semantic concept tags of documents can therefore serve to associate documents related in unseen contexts, or categorize them, in information filtering or retrieval scenarios. Thus, inspired by the success of sequence-to-sequence neural models, we develop a novel *sequence-to-set* framework with attention, for learning document representations in a unique unsupervised setting, using no human-annotated document labels or external knowledge resources and only corpus-derived term statistics to drive the training. This can effect term transfer within a corpus for semantically tagging a large collection of documents. Our sequence-to-set modeling approach to predict semantic tags , gives to the best of our knowledge, the state-of-the-art for both, an **unsupervised** query expansion (QE) task for the **TREC CDS 2016** challenge dataset when evaluated on an Okapi BM25–based document retrieval system; and also over the MLTM system baseline (Soleimani and Miller, 2016), for both **supervised** and **semi-supervised** multi-label prediction tasks with **del.icio.us** and **Ohsumed** datasets. We make our code and data publicly available[1].

1 Introduction

Recent times have seen an upsurge in efforts towards personalized medicine where clinicians tailor their medical decisions to the individual patient, based on the patient's genetic information, other molecular analysis, and the patient's preference. This often requires them to combine clinical experience with evidence from scientific research, such as that available from biomedical literature, in a process known as evidence-based medicine (EBM). Finding the most relevant recent research however, is challenging not only due to the volume and the pace at which new research is being published, but also due to the complex nature of the information need, arising for example, out of a clinical note which may be used as a query. This calls for better automated methods for natural language understanding (NLU), e.g. to derive a set of key terms or *related concepts* helpful in appropriately transforming a complex query, by reformulation so as to be able to handle and possibly resolve medical jargon, lesser-used acronyms, misspelling, multiple subject areas and often multiple references to the same entity or concept, and retrieve the most related, yet most comprehensive set of useful results.

At the same time, tremendous strides have been made by recent neural machine learning models in reasoning with texts on a wide variety of NLP tasks. In particular, sequence-to-sequence (seq2seq) neural models often employing *attention* mechanisms, have been largely successful in delivering the state-of-the-art for tasks such as machine translation (Bahdanau et al., 2014), (Vaswani et al., 2017), handwriting synthesis (Graves, 2013), image captioning (Xu et al., 2015), speech recognition (Chorowski et al., 2015) and document summarization (Cheng and Lapata, 2016). Inspired by these successes, we aimed to harness the power of sequential *encoder-decoder* architectures with attention, to train end-to-end differentiable models that are able to learn the best possible representation of input documents in a collection while being predictive of a set of *key terms* that best describe the docu-

[1] https://github.com/mcoqzeug/seq2set-semantic-tagging

Proceedings of the BioNLP 2020 workshop, pages 14–27

ment. These will be later used to *transfer* a relevant but diverse set of key terms from the most related documents, for "semantic tagging" of the original input documents so as to aid in downstream query refinement for IR by pseudo-relevance feedback (Xu and Croft, 2000).

To this end and to the best of our knowledge, we are the first to employ a novel, completely unsupervised end-to-end neural attention-based document representation learning approach, using no external labels, in order to achieve the most meaningful term transfer between related documents, i.e. semantic tagging of documents, in a "pseudo-relevance feedback"–based (Xu and Croft, 2000) setting for unsupervised query expansion. This may also be seen as a method of document expansion as a means for obtaining query refinement terms for downstream IR. The following sections give an account of our specific architectural considerations in achieving an end-to-end neural framework for semantic tagging of documents using their representations, and a discussion of the results obtained from this approach.

2 Related Work

Pseudo-relevance feedback (PRF), a *local context analysis* method for automatic query expansion (QE), is extensively studied in information retrieval (IR) research as a means of addressing the word mismatch between queries and documents. It adjusts a query relative to the documents that initially appear to match it, with the main assumption that the top-ranked documents in the first retrieval result contain many useful terms that can help discriminate relevant documents from irrelevant ones (Xu and Croft, 2000), (Cao et al., 2008). It is motivated by *relevance feedback* (RF), a well-known IR technique that modifies a query based on the relevance judgments of the retrieved documents (Salton et al., 1990). It typically adds common terms from the relevant documents to a query and re-weights the expanded query based on term frequencies in the relevant documents relative to the non-relevant ones. Thus in PRF we find an initial set of most relevant documents, then assuming that the top k ranked documents are relevant, RF is done as before, without manual interaction by the user. The added terms are, therefore, common terms from the top-ranked documents.

To this end, (Cao et al., 2008) employ term classification for retrieval effectiveness, in a "supervised" setting, to select most relevant terms. (Palangi et al., 2016) employ a deep sentence embedding approach using LSTMs and show improvement over standard sentence embedding methods, but as a means for directly deriving encodings of queries and documents for use in IR, and not as a method for QE by PRF. In another approach, (Xu et al., 2017) train autoencoder representations of queries and documents to enrich the feature space for learning-to-rank, and show gains in retrieval performance over pre-trained rankers. But this is a fully supervised setup where the queries are *seen* at train time. (Pfeiffer et al., 2018) also use an autoencoder-based approach for actual query refinement in pharmacogenomic document retrieval. However here too, their document ranking model uses the encoding of the query and the document for training the ranker, hence the queries are *not unseen* with respect to the document during training. They mention that their work can be improved upon by the use of seq2seq-based approaches. In this sense, i.e. with respect to QE by PRF and learning a sequential document representation for document ranking, our work is most similar to (Pfeiffer et al., 2018). However the queries are completely unseen in our case and we use only the documents in the corpus, to train our neural document language models from scratch in a completely unsupervised way.

Classic sequence-to-sequence models like (Sutskever et al., 2014) demonstrate the strength of recurrent models such as the LSTM in capturing short and long range dependencies in learning effective encodings for the end task. Works such as (Graves, 2013), (Bahdanau et al., 2014), (Rocktäschel et al., 2015), further stress the key role that attention, and multi-headed attention (Vaswani et al., 2017) can play in solving the end task. We use these insights in our work.

According to the detailed report provided for this dataset and task in (Roberts et al., 2016) all of the systems described perform **direct query reweighting** aside from **supervised term expansion** and are highly tuned to the clinical queris in this dataset. In a related medical IR challenge (Roberts et al., 2017) the authors specifically mention that with only six partially annotated queries for system development, it is likely that systems were either under- or over-tuned on these queries. Since the setup of the seq2set framework is an attempt to model the PRF based query expansion method of its closest related work (Das et al., 2018) where

the effort is also to train a neural generalized language model for unsupervised semantic tagging, we choose this system as the benchmark to compare against to our end-to-end approach for the same task.

3 Methodology

Drawing on sequence-to-sequence modeling approaches for text classification, e.g. textual entailment (Rocktäschel et al., 2015) and machine translation (Sutskever et al., 2014), (Bahdanau et al., 2014) we adapt from these settings into a *sequence-to-set* framework, for learning representations of input documents, in order to derive a meaningful set of terms, or *semantic tags* drawn from a closely related set of documents, that expand the original documents. These document expansion terms are then used downstream for query reformulation via PRF, for unseen queries. We employ an end-to-end framework for unsupervised representation learning of documents using TFIDF-based *pseudo-labels* (Figure 1(a))and a separate cosine similarity-based ranking module for semantic tag inference (Figure 1(b)).

We employ various methods such as doc2vec, Deep Averaging, sequential models such as LSTM, GRU, BiGRU, BiLSTM, BiLSTM with Attention and Self-attention, detailed in Figure 1(c)-(f), see Appendix A, for learning fixed-length input document representations in our framework. We apply methods like DAN (Iyyer et al., 2015), LSTM, and BiLSTM as our baselines and formulate attentional models including a self-attentional Transformer-based one (Vaswani et al., 2017) as our proposed augmented document encoders.

Further, we hypothesize that a sequential, bi-directional or attentional encoder coupled with a decoder, i.e. a sigmoid or softmax prediction layer, that conditions on the encoder output v (similar to an approach by (Kiros et al., 2015) for learning a neural probabilistic language model), would enable learning of the optimal semantic tags in our unsupervised query expansion setting, while modeling directly for this task in an end-to-end neural framework. In our setup the decoder predicts a meaningful set of concept tags that best describe a document according to the training objective. The following sections describe our setup.

3.1 The *Sequence-to-Set* Semantic Tagging Framework

Task Definition: For each query document d_q in a given a collection of documents $D = \{d_1, d_2, ..., d_N\}$, represented by a set of k keywords or labels, e.g. k terms in d_q derived from $top-|V|$ TFIDF-scored terms, find an ***alternate*** set of k most relevant terms coming from documents *"most related"* to d_q from elsewhere in the collection. These serve as *semantic tags* for expanding d_q.

In the **unsupervised** task setting described later, a document to be tagged is regarded as a query document d_q; its semantic tags are generated via PRF, and these terms will in turn be used for PRF–based expansion of unseen queries in downstream IR. Thus d_q could represent an original complex query text or a document in the collection.

In the following sections we describe the building blocks used in the setup for the baseline and proposed models for sequence-to-set semantic tagging as described in the task definition.

3.2 Training and Inference Setup

The overall architecture for sequence-to-set semantic tagging consists of two phases, as depicted in the block diagrams in Figures 1(a) and 1(b): the first, for training of input representations of documents; and the second for inference to achieve *term transfer* for semantic tagging. As shown in Figure 1(a), the proposed model architecture would first learn the appropriate feature representations of documents in a first pass of training, by taking in the tokens of an input document sequentially, using a document's pre-determined $top-k$ TFIDF-scored terms as the ***pseudo-***class labels for an input instance, i.e. prediction targets for a $sigmoid$ layer for multi-label classification. The training objective is to maximize probability for these k terms, or $y_p = (t_1, t_2, ..t_k) \in V$, i.e.

$$\arg\max_{\theta} P(y_p = (t_1, t_2, ..t_k) \in V | v; \theta) \quad (1)$$

given the document's encoding v. For computational efficiency, we take V to be the list of top-10K TFIDF-scored terms from our corpus, thus $|V| = 10,000$. k is taken as 3, so each document is initially labeled with 3 terms. The sequential model is then trained with the k-hot 10K-dimensional label vector as targets for the sigmoid classification layer, employing a couple of alternative training

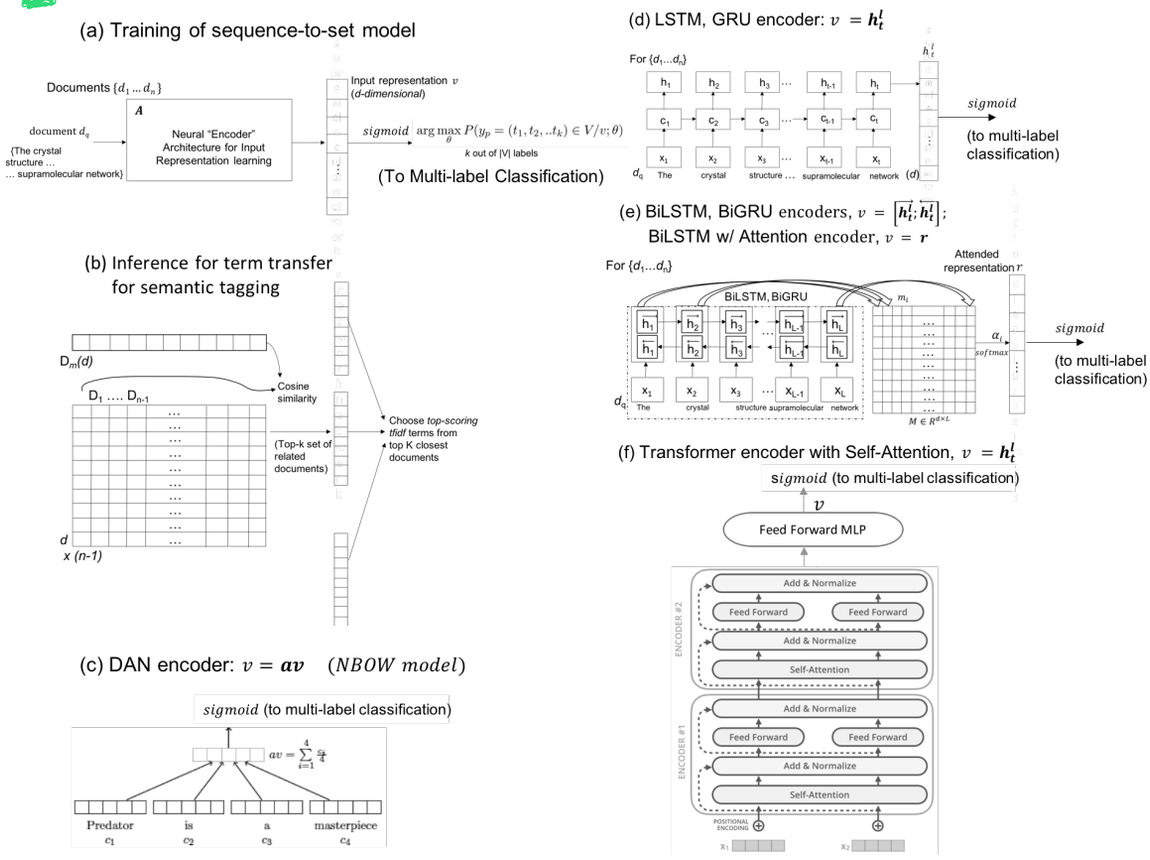

Figure 1: Overview of *Sequence-to-Set* Framework. (a) Method for training document or query representations, (b) Method for Inference via **term transfer** for semantic tagging; Document Sequence Encoders: (c) Deep Averaging encoder; (d) LSTM last hidden state, GRU encoders; (e) BiLSTM last hidden state, BiGRU (shown in dotted box), BiLSTM attended hidden states encoders; and (f) Transformer self-attentional encoder [source: (Alammar, 2018)].

objectives. The first, typical for multi-label classification, minimizes a categorical cross-entropy loss, which for a single training instance with ground-truth label *set*, y_p, is:

$$L_{CE}(\hat{y_p}) = \sum_{i=1}^{|V|} y_i \log(\hat{y_i}) \tag{2}$$

Since our goal is to obtain the most meaningful document representations most predictive of their assigned terms, and that can also be predictive of semantic tags not present in the document, we also consider a language model–based loss objective converting our decoder to a neural language model. Thus, we employ a training objective that maximizes the conditional log likelihood of the label terms L_d of a document d_q, given the document's representation v, i.e. $P(L_d|d_q)$ (where $y_p = L_d \in V$). This amounts to minimizing the negative log likelihood of the label representations conditioned on the document encoding. Thus,

$$P(L_d|d_q) = \prod_{l \in L_d} P(l|d_q) = -\sum_{l \in L_d} \log(P(l|d_q)) \tag{3}$$

Since $P(l|d_q) \propto \exp(v_l \cdot v)$, where v_l and v are the label and document encodings, it is equivalent to minimizing:

$$L_{LM}(\hat{y_p}) = -\sum_{l \in L_d} \log(\exp(v_l \cdot v)) \tag{4}$$

Equation (4) represents our language model–style loss objective. We run experiments training with both losses (Equations (2) & (4)) as well as a variant that is a summation of both, with a hyper-parameter α used to tune the language model component of the total loss objective.

4 Task Settings

4.1 Unsupervised Task – Semantic Tagging for Query Expansion

We now describe the setup and results for experiments run on our unsupervised task setting of semantic tagging of documents for PRF–based query expansion.

4.1.1 Dataset – TREC CDS 2016

The 2016 TREC CDS challenge dataset, makes available actual electronic health records (EHR) of patients (de-identified), in the form of case reports, typically describing a challenging medical case. Such a case report represents a **query** in our system, having a complex information need. There are 30 queries in this dataset, corresponding to such case reports, at 3 levels of granularity **Note**, **Description** and **Summary** text as described in (Roberts et al., 2016). The target document collection is the Open Access Subset of PubMed Central (PMC), containing 1.25 million articles consisting of *title*, *keywords*, *abstract* and *body* sections. We use a subset of 100K of these articles for which human relevance judgments are made available by TREC, for training. Final evaluation however is done on an ElasticSearch index built on top of the entire collection of 1.25 million PMC articles.

4.1.2 Unsupervised Task Experiments

We ran several sets of experiments with various document encoders, employing pre-trained and fine-tuned embedding schemes like skip-gram (Mikolov et al., 2013a) and Probabilistic FastText (Athiwaratkun et al., 2018), see Appendix B. The experimental setup used is the same as the Phrase2VecGLM (Das et al., 2018), the only other known system for this dataset, that performs "unsupervised semantic tagging of documents by PRF", for downstream query expansion. Thus we take this system as the current state-of-the-art system baseline, while our *non-attention-based* document encoding models constitute our standard baselines. Our document-TFIDF representations–based query expansion forms yet another baseline. Summary text UMLS (Lindberg et al., 1993; Bodenreider, 2004) terms for use in our augmented models is available to us via the UMLS Java Metamap API (Demner-Fushman et al., 2017). The first was a set of experiments with our different models using the `Summary Text` as the base query. Following this we ran experiments with our models using the `Summary Text + Sum. UMLS terms` as the "augmented" query. We use the Adam optimizer (Kingma and Ba, 2014) for training our models. After several rounds of hyper-paramater tuning, $batch\_size$ was set to 128, $dropout$ to 0.3, the prediction layer was fixed to $sigmoid$, the loss function switched between cross-entropy and summation of cross entropy and LM losses, and models trained with early stopping.

Results from various *Seq2Set* encoder models on **base** (`Summary Text`) and **augmented** (`Summary Text + Summary-based UMLS terms`) query, are outlined in Table 1. Evaluating on base query, a *Seq2Set*-Transformer model beats all other *Seq2Set* encoders, and also the TFIDF, MeSH QE terms and Expert QE terms baselines. On the augmented query, the *Seq2Set*-BiGRU and *Seq2Set*-Transformer models outperform all other *Seq2Set* encoders, and *Seq2Set*-Transformer outperforms all non-ensemble baselines and the Phrase2VecGLM unsupervised QE ensemble system baseline significantly, with P@10 of **0.4333**. Best performing *supervised QE* systems for this dataset, tuned on all 30 queries, range between *0.35–0.4033* P@10 (Roberts et al., 2016), better than unsupervised QE systems on base query, but surpassed by the best *Seq2Set*-based models such as *Seq2Set*-Transformer on augmented query, even without ensemble. Semantic tags from a best-performing model, do appear to pick terms relating to certain conditions, e.g.: <***query_doc original pseudo-label terms***:*['obesity', 'diabetes', 'pulmonary-hypertension', 'children']*, ***semantic tags***: *['dyslipidaemia', 'hyperglycemia', 'bmi', 'subjects']* >.

4.2 Supervised Task – Automated Text Categorization

The Seq2set framework's unsupervised semantic tagging setup is primarily applicable in those settings where no pre-existing document labels are available. In such a scenario, of unsupervised semantic tagging of a large document collection, the Seq2set framework therefore consists of separate training and inference steps to infer tags from other documents after encodings have been learnt. We therefore conduct a series of extensive evaluations in the manner described in the previous section, using a downstream QE task in order to validate our method. However, when a tagged document

Unsupervised QE Systems (Base Query)	**P@10**
BM25+*Seq2Set*-doc2vec (**baseline**)	0.0794
BM25+*Seq2Set*-TFIDF Terms (**baseline**)	0.2000
BM25+**MeSH** QE Terms (**baseline**)	0.2294
BM25+**Human Expert** QE Terms (**baseline**)	0.2511
BM25+unigramGLM+Phrase2VecGLM *ensemble* (**system baseline**)	**0.2756**
BM25+*Seq2Set*-Transformer (L_{CE}) (**model**)	**0.2861***
Supervised QE Systems (Base Query)	
BM25+ETH Zurich-*ETHSummRR*	0.3067
BM25+Fudan Univ.DMIIP-*AutoSummary1*	**0.4033**
Unsupervised QE Systems (Augmented Query)	
BM25+*Seq2Set*-doc2vec (**baseline**)	0.1345
BM25+*Seq2Set*-TFIDF Terms (**baseline**)	0.3000
BM25+unigramGLM+Phrase2VecGLM *ensemble* (**system baseline**)	**0.3091**
BM25+*Seq2Set*-BiGRU (LM only loss) (**model**)	**0.3333***
BM25+*Seq2Set*-Transformer ($L_{CE} + L_{LM}$) (**model**)	**0.4333***

Table 1: Results on IR for best *Seq2set* models, in an ***unsupervised PRF***–based QE setting. Boldface indicates statistical significance @p<<0.01 over previous.

collection is available where the set of document labels are already known, we can learn to predict tags from this set of known labels on a new set of similar documents. Thus, in order to generalize our Seq2set approach to such other tasks and setups, we therefore aim to validate the performance of our framework on such a labeled dataset of tagged documents, which is equivalent to adapting the Seq2set framework for a supervised setup. In this setup we therefore only need to use the training module of the Seq2set framework shown in Figure 1(a), and measure tag prediction performance on a held out set of documents. For this evaluation, we therefore choose to work with the popular Delicious (del.icio.us) folksonomy dataset, same as that used by (Soleimani and Miller, 2016) in order to do an appropriate comparison with their MLTM framework that is also evaluated on a similar document multi-label prediction task.

4.2.1 Dataset – del.icio.us

The Delicious dataset contains tagged web pages retrieved from the social bookmarking site, del.icio.us. There are 20 common tags used as class labels: *reference*, *design*, *programming*, *internet*, *computer*, *web*, *java*, *writing*, *English*, *grammar*, *style*, *language*, *books*, *education*, *philosophy*, *politics*, *religion*, *science*, *history* and *culture*. The training set consists of 8250 documents and the test set consists of 4000 documents.

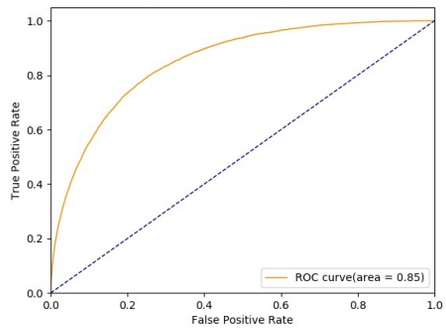

Figure 2: Seq2Set–**supervised** on **del.icio.us**, best **Transformer** model–based encoder

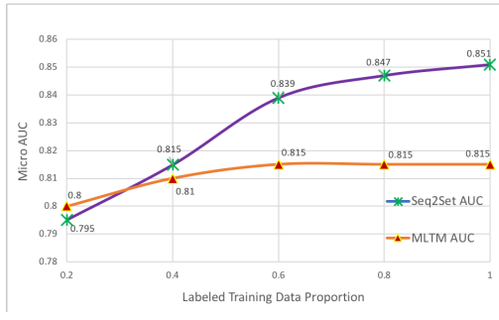

Figure 3: A comparison of document labeling performance of Seq2set versus MLTM

4.2.2 Supervised Task Experiments

We then run Seq2set-based training for our 8 different encoder models on the training set for the 20 labels, and perform evaluation on the test set measuring sentence-level ROC AUC on the labeled documents in the test set.

Figure 2 shows the ROC AUC for the best performing Transformer model from the Seq2set framework on the del.icio.us dataset, which was trained with a sigmoid–based prediction layer on cross entropy loss with a batch size of 64 and dropout set to 0.3. This best model got an ROC AUC of **0.85** , statistically significantly surpassing MLTM (AUC 0.81 @ $p << 0.001$) for this task and dataset.

Figure 3 also shows a comparison of the ROC AUC scores obtained with training Seq2set and MLTM based models for this task with various labeled data proportions. Here again we see that Seq2set has clear advantage over the current MLTM state-of-the-art, statistically significantly surpassing it ($p << 0.01$) when trained with greater than 25% of the labeled dataset.

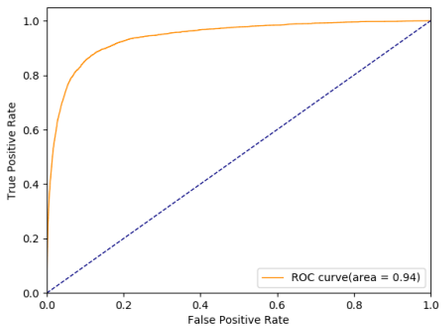

Figure 4: Seq2Set–**semi-supervised** on **Ohsumed**, best **Transformer** model–based encoder w/ Cross Entropy–based Softmax prediction; 4 layers, 10 attention heads, dropout=0

4.3 Semi-Supervised Text Categorization

We then seek to further validate how well the Seq2set framework can leverage large scale pre-training on unlabeled data given only a small amount of labeled data for training, to be able to improve prediction performance on a held out set of these known labels. This amounts to a semi-supervised setup–based evaluation of the Seq2set framework. In this setup, we perform the training and evaluation of Seq2set similar to the supervised setup, except we have an added step of pre-training the multi-label prediction on large amounts of unlabeled document data in exactly the same way as the unsupervised setup.

4.3.1 Dataset – Ohsumed

We employ the Ohsumed dataset available from the TREC Information Filtering tracks of years 87-91 and the version of the labeled **Ohsumed** dataset used by (Soleimani and Miller, 2016) for evaluation, to have an appropriate comparison with their MLTM system also evaluated for this dataset. The version of the Ohsumed dataset due to (Soleimani and Miller, 2016) consists of 11122 training and 5388 test documents, each assigned to one or multiple labels of 23 MeSH diseases categories. Almost half of the documents have more than one label.

4.3.2 Semi-Supervised Task Experiments

We first train and test our framework on the labeled subset of the **Ohsumed** data from (Soleimani and Miller, 2016) similar to the supervised setup described in the previous section. This evaluation gives a statistically significant ROC AUC of 0.93 over the 0.90 AUC for the MLTM system of (Soleimani and Miller, 2016) for a Transformer–based Seq2set model performing best. Next we experiment with the semi-supervised setting where we first train the Seq2set framework models on a large number of documents that do not have pre-existing labels. This pre-training is performed in exactly a similar fashion as the unsupervised setup. Thus we first preprocess the Ohsumed data from years 87-90 to obtain a top-1000 TFIDF score–based vocabulary of tags, pseudo-labeling all the documents in the training set with these. Our training and evaluation for the semi-supervised setup consists of 3 phases: **Phase 1**: We employ our seq2set framework (using each one of our encoder models) for multi-label prediction on this pseudo-labeled data, having an output prediction layer of 1000 having a penultimate fully-connected layer of dimension 23, same as the number of labels in the Ohsumed dataset; **Phase 2**: After pre-training with pseudolabels we discard the final layer and continue to train labeled Ohsumed dataset from 91 by 5-fold cross-validation with early stopping. **Phase 3**: This is the final evaluation step of our semi-supervised trained Seq2set model on the labeled Ohsumed test dataset used by (Soleimani and Miller, 2016). This constitutes simply inferring predicted tags using the trained model on the test data. As shown in Figure 4, our evaluation of the Seq2set framework for the Ohsumed dataset, comparing supervised and semi-supervised training setups, yields an ROC AUC of **0.94** for our best performing **semi-supervised**–trained model of Fig. 4, compared to the various supervised trained models for the same dataset that got a best ROC AUC of 0.93. The top performing semi-supervised model again involves a Transformer–based encoder using a softmax layer for prediction, with 4 layers, 10 attention heads, and no dropout. Thus, the best results on the semi-supervised training experiments (ROC AUC 0.94) **statistically significantly outperforms** ($p << 0.01$) the **MLTM** system baseline (ROC AUC 0.90) on the Ohsumed dataset, while also clearly surpassing the top-performing supervised Seq2set models on the same dataset. This demonstrates that our *Seq2set* framework is able to leverage the benefits of data augmentation in the semi-supervised setup by training with large amounts of unlabeled data on top of limited labeled data.

5 Conclusion

We develop a novel *sequence-to-set* end-to-end encoder-decoder–based neural framework for multi-label prediction, by training document representations using no external supervision labels, for pseudo-relevance feedback–based unsupervised semantic tagging of a large collection of documents. We find that in this unsupervised task setting of PRF-based semantic tagging for query expansion, a multi-term prediction training objective that jointly optimizes both prediction of the TFIDF–based document pseudo-labels and the log likelihood of the labels given the document encoding, surpasses previous methods such as Phrase2VecGLM (Das et al., 2018) that used neural generalized language models for the same. Our initial hypothesis that bi-directional or self-attentional models could learn the most efficient semantic representations of documents when coupled with a loss more effective than cross-entropy at reducing language model perplexity of document encodings, is corroborated in all experimental setups. We demonstrate the effectiveness of our novel framework in every task setting, viz. for **unsupervised** QE via PRF-based semantic tagging for a downstream medical IR challenge task; as well as for both, **supervised** and **semi-supervised** task settings, where Seq2set statistically significantly outperforms the state-of-art MLTM baseline (Soleimani and Miller, 2016) on the same held out set of documents as MLTM, for multi-label prediction on a set of known labels, for automated text categorization; achieving to the best of our knowledge, the current state-of-the-art for multi-label prediction on documents, with or without known labels. We therefore demonstrate the effectiveness of our Sequence-to-Set framework for multi-label prediction, on any set of documents, applicable especially towards the automated categorization, filtering and semantic tagging for QE-based retrieval, of biomedical literature for EBM. Future directions would involve experiments replacing TDIDF labels with more meaningful terms (using unsupervised term extraction) for query expansion, initialization with pre-trained embeddings for the biomedical domain such as BlueBERT (Peng et al., 2019) and BioBERT (Lee et al., 2020), and multi-task learning with closely related tasks such as biomedical Named Entity Recognition and Relation Extraction to learn better document representations and thus more meaningful semantic tags of documents useful for downstream EBM tasks.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Jay Alammar. 2018. The illustrated transformer. Online; posted June 27, 2018.

Ben Athiwaratkun, Andrew Gordon Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250. ACM.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075.

Manirupa Das, Eric Fosler-Lussier, Simon Lin, Soheil Moosavinasab, David Chen, Steve Rust, Yungui Huang, and Rajiv Ramnath. 2018. Phrase2vecglm: Neural generalized language model–based semantic tagging for complex query reformulation in medical ir. In *Proceedings of the BioNLP 2018 workshop*, pages 118–128.

Dina Demner-Fushman, Willie J Rogers, and Alan R Aronson. 2017. Metamap lite: an evaluation of a new java implementation of metamap. *Journal*

of the American Medical Informatics Association, 24(4):841–844.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the ACL*, volume 1, pages 1681–1691.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Donald AB Lindberg, Betsy L Humphreys, and Alexa T McCray. 1993. The unified medical language system. *Yearbook of Medical Informatics*, 2(01):41–51.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):694–707.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Jonas Pfeiffer, Samuel Broscheit, Rainer Gemulla, and Mathias Göschl. 2018. A neural autoencoder approach for document ranking and query refinement in pharmacogenomic information retrieval. In *Proceedings of the BioNLP 2018 workshop*, pages 87–97.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Kirk Roberts, Anupama E Gururaj, Xiaoling Chen, Saeid Pournejati, William R Hersh, Dina Demner-Fushman, Lucila Ohno-Machado, Trevor Cohen, and Hua Xu. 2017. Information retrieval for biomedical datasets: the 2016 biocaddie dataset retrieval challenge. *Database*, 2017.

Kirk Roberts, Ellen Voorhees, Dina Demner-Fushman, and William R. Hersh. 2016. Overview of the trec 2016 clinical decision support track. Online; posted August-2016.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Gerard Salton, Chris Buckley, and Maria Smith. 1990. On the application of syntactic methodologies in automatic text analysis. *Information Processing & Management*, 26(1):73–92.

Hossein Soleimani and David J Miller. 2016. Semi-supervised multi-label topic models for document classification and sentence labeling. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 105–114. ACM.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Bo Xu, Hongfei Lin, Yuan Lin, and Kan Xu. 2017. Learning to rank with query-level semi-supervised autoencoders. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2395–2398. ACM.

Jinxi Xu and W Bruce Croft. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)*, 18(1):79–112.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

A Sequence-based Document Encoders

We describe below the different neural models that we use for the sequence encoder, as part of our encoder-decoder architecture for deriving semantic tags for documents.

A.1 doc2vec encoder

$doc2vec$ is the unsupervised algorithm due to (Le and Mikolov, 2014), that learns fixed-length representations of variable length documents, representing each document by a dense vector trained to predict surrounding words in contexts sampled from each document. We derive these doc2vec encodings by pre-training on our corpus. We then use them directly as features for inferring semantic tags per Figure 1(b) without training them within our framework against the loss objectives. We expect this to be a strong document encoding baseline in capturing the semantics of documents. TFIDF Terms is our other baseline where we don't train within the framework but rather use the top-k neighbor documents' TFIDF pseudo-labels as the semantic tags for the query document.

A.2 Deep Averaging Network encoder

The Deep Averaging Network (DAN) for text classification due to (Iyyer et al., 2015) Figure 1 (c), is formulated as a neural bag of words encoder model for mapping an input sequence of tokens X to *one* of k labels. v is the output of a composition function g, in this case $averaging$, applied to the sequence of word embeddings v_w for $w \in X$. For our multi-label classification problem, v is fed to a *sigmoid* layer to obtain scores for each independent classification. We expect this to be another strong document encoder given results in the literature and it proves in practice to be.

A.2.1 LSTM and BiLSTM encoders

LSTMs (Hochreiter and Schmidhuber, 1997), by design, encompass memory cells that can store information for a long period of time and are therefore capable of learning and remembering over long and variable sequences of inputs. In addition to three types of gates, i.e. *input*, *forget*, and *output* gates, that control the flow of information into and out of these cells, LSTMs have a hidden state vector h_t^l, and a memory vector c_t^l. At each time step, corresponding to a token of the input document, the LSTM can choose to read from, write to, or reset the cell using explicit gating mechanisms. Thus the LSTM is able to learn a language model for the entire document, encoded in the hidden state of the final timestep, which we use as the document encoding to give to the prediction layer. By the same token, owing to the bi-directional processing of its input, a BiLSTM-based document representation is expected to be even more robust at capturing document semantics than the LSTM, with respect to its prediction targets. Here, the document representation used for final classification is the concatenated hidden state outputs from the final step, $[\overrightarrow{h}_t^l; \overleftarrow{h}_t^l]$, depicted by the dotted box in Fig. 1(e).

A.3 BiLSTM with Attention encoder

In addition, we also propose a BiLSTM with attention-based document encoder, where the output representation is the *weighted combination* of the concatenated hidden states at each time step. Thus we learn an *attention-weighted* representation at the final output as follows. Let $X \in^{(d \times L)}$ be a matrix consisting of output vectors $[h_1, \ldots, h_L]$ that the Bi-LSTM produces when reading L tokens of the input document. Each word representation h_i is obtained by concatenating the forward and backward hidden states, i.e. $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$. d is the size of embeddings and hidden layers. The attention mechanism produces a vector α of attention weights and a weighted representation r of the input, via:

$$M = \tanh(WX), \quad M \in^{(d \times L)} \tag{5}$$

$$\alpha = \text{softmax}(w^T M), \quad \alpha \in^{L} \tag{6}$$

$$r = X\alpha^T, \quad r \in^{d} \tag{7}$$

Here, the intermediate attention representation m_i (i.e. the i^{th} column vector in M) of the i^{th} word in the input document is obtained by applying a non-linearity on the matrix of output vectors X, and the attention weight for the i^{th} word in the input is the result of a weighted combination (parameterized by w) of values in m_i. Thus $r \in^{d}$ is the $attention-$weighted representation of the word and phrase tokens in an input document used in optimizing the training objective in downstream multi-label classification, as shown by the final attended representation r in Figure 1(e).

A.4 GRU and BiGRU encoders

A Gated Recurrent Unit (GRU) is a type of recurrent unit in recurrent neural networks (RNNs)

that aims at tracking long-term dependencies while keeping the gradients in a reasonable range. In contrast to the LSTM, a GRU has only 2 gates: a reset gate and an update gate. First proposed by (Chung et al., 2014), (Chung et al., 2015) to make each recurrent unit to adaptively capture dependencies of different time scales, the GRU, however, does not have any mechanism to control the degree to which its state is exposed, exposing the whole state each time. In the LSTM unit, the amount of the memory content that is seen, or used by other units in the network is controlled by the output gate, while the GRU exposes its full content without any control. Since the GRU has simpler structure, models using GRUs generally converge faster than LSTMs, hence they are faster to train and may give better performance in some cases for sequence modeling tasks. The BiGRU has the same structure as GRU except constructed for bi-directional processing of the input, depicted by the dotted box in Fig. 1(e).

A.5 Transformer self-attentional encoder

Recently, the Transformer encoder-decoder architecture due to (Vaswani et al., 2017), based on a *self-attention* mechanism in the encoder and decoder, has achieved the state-of-the-art in machine translation tasks at a fraction of the computation cost. Based entirely on attention, and replacing the recurrent layers commonly used in encoder-decoder architectures with *multi-headed* self-attention, it has outperformed most previously reported ensembles on the task. Thus we hypothesize that this self-attention-based model could learn the most efficient semantic representations of documents for our unsupervised task. Since our models use $tensorflow$ (Abadi et al., 2016), a natural choice was document representation learning using the Transformer model's available $tensor2tensor$ API. We hoped to leverage apart from the computational advantages of this model, the capability of capturing semantics over varying lengths of context in the input document, afforded by multi-headed self-attention, Figure 1(f). Self-attention is realized in this architecture, by training 3 matrices, made up of vectors, corresponding to a Query vector, a Key Vector and a Value vector for each token in the input sequence. The output of each self-attention layer is a summation of weighted Value vectors that passes on to a feed-forward neural network. Position-based encoding to replace recurrences help to lend more parallelism to computations and make things faster. *Multi-headed* self-attention further lends the model the ability to focus on different positions in the input, with multiple sets of Query/Key/Value weight matrices, which we hypothesize should result in the most effective document representation, among all the models, for our downstream task.

A.6 CNN encoder

Inspired by the success of (Kim, 2014) in employing CNN architectures successfully for achieving gains in NLP tasks we also employ a CNN-based encoder in the seq2set framework. (Kim, 2014) train a simple CNN with a layer of convolution on top of pre-trained word vectors, as a sequence of length n embeddings concatenated to form a matrix input. Filters of different sizes, representing various context windows over neighboring words, are then applied to this input, over each possible window of words in the sequence to obtain feature maps. This is followed by a max-over-time pooling operation to take maximum value of the feature map as the feature corresponding to a particular filter. The model then combines these features to form a penultimate layer which is passed to a fully connected softmax layer whose output is the probability distribution over labels. In case of seq2set these features are passed to sigmoid layer for final multi-label prediction used cross entropy loss or a combination of cross-entropy and LM losses. We use filters of sizes 2, 3, 4 and 5. Like our other encoders, we fine-tune the document representations learnt.

B Embedding Algorithms Experimented with

We describe here the various algorithms used to train word embeddings for use in our models.

Skip-Gram word2vec: We generate word embeddings trained with the skip-gram model with negative sampling (Mikolov et al., 2013b) with dimension settings of 50 with a context window of 4, and also 300, with a context window of 5, using the $gensim$ package [2] (Řehůřek and Sojka, 2010).

Probabilistic FastText: The Probabilistic FastText (PFT) word embedding model of (Athiwaratkun et al., 2018) represents each word with a Gaussian mixture density, where the mean of a mixture component given by the sum of n-grams, can capture multiple word senses, sub-word structure,

[2] https://radimrehurek.com/gensim/

and uncertainty information. This model outperforms the n-gram averaging of FastText getting state-of-the-art performance on several word similarity and disambiguation benchmarks. The probabilistic word representations with flexible sub-word structures, can achieve multi-sense representations that also give rich semantics for rare words. This makes them very suitable to generalize for rare and out-of-vocabulary words motivating us to opt for PFT-based word vector pre-training [3] over regular FastText.

ELMo: Another consideration was to use embeddings that can explicitly capture the language model underlying sentences within a document. ELMo (Embeddings from Language Models) word vectors (Peters et al., 2018) presented such a choice where the vectors are derived from a bidirectional LSTM trained with a coupled language model (LM) objective on a large text corpus. The representations are a function of all of the internal layers of the biLM. Using linear combinations of the vectors derived from each internal state has shown marked improvements over various downstream NLP tasks, because the higher-level LSTM states capture context-dependent aspects of word meaning (e.g., they can be used without modification to perform well on supervised word sense disambiguation tasks) while lower- level states model aspects of syntax. Using the API [4] we generate ELMo embeddings fine-tuned for our corpus with dimension settings of 50 and 100 using only the top layer final representations. A discussion of the results from each set of experiments is outlined in the following section and summarized in Table 1.

C Experimental Considerations and Hyperparameter Settings

Of the metrics available, P@10 gives the number of relevant items returned in the top-10 results and NDCG looks at precision of the returned items at the correct rankings. For our particular dataset domain, the number of relevant results returned in the top-10 is more important, hence Table 1 reports results ranked in ascending order of P@10.

The PRF setting shown in the results table means that, we take the top 10-15 documents returned by an ElasticSearch (ES) index for each of the 30 Summary Text queries in our dataset, and subsequently use the semantic tags assigned to each of these top documents as the terms for query expansion for the original query. We then re-run these expanded queries through the ES index to record the retrieval performance. Thus the queries our system is evaluated on, are *not seen* at the time of training our models, but only during evaluation, hence it is *unsupervised QE*.

Similar to Das et al. (2018), for the feedback loop based query expansion method, we had two separate human judgment–based baselines, one using the MeSH terms available from PMC for the top 15 documents returned in a first round of querying with Summary text, and the other based on human expert annotations of the 30 query topics, made available by the authors.

Since we had mixed results initially with our models, we explored various options to increase the training signal. First was by the use of $neighbor$-document's labels in our label vector for training. In this scheme, we used the 3-hot TFIDF label representation for each document to pick a list of n–nearest neighbors to it. We then included the labels of those nearest documents into the label vector for the original document for use during training. We experimented with choices $3, 5, 10, 15, 20, 25$ for n. We observed improvements with incorporating neighbor labels.

Next we experimented with incorporating word neighbors into the input sequence for our models. We did this in two different ways, the first was to average all the neighbors and concatenate with the original token embedding, the other was to average all of the embeddings together. The word itself was always weighted more than the neighbors. This scheme also gave improvement.

Finally we experimented with incorporating embeddings pre-trained by latest state-of-the-art methods (Appendix B) as the input tokens for our models. After several rounds of hyper-parameter tuning, $batch\_size$ was set to 128, and $dropout$ to 0.3. We also performed a small grid search into the space of hyperparameters like number of hidden layers varied as $2, 3, 4$, and α varied as $[1.0, 10.0, 100.0, 1000.0, 10000.0]$, determining the best settings for each encoder.

A glossary of acronyms and parameters used in training of our models is as follows: sg=skip-gram; pft=Probablistic FastText; $elmo$=ELMo; d=embedding dimension; kln=number of "neighbor documents'" labels; nl=number of hidden layers in the model; h= Number of multi-attention

[3] https://github.com/benathi/multisense-prob-fasttext
[4] https://allennlp.org/elmo

heads; bs=batch size; dp=dropout; ep=no. of epochs; α=weight parameter for language model loss component.

Best-performing model settings: Our best performing models on the base query was a Transformer encoder with 10 attention heads: $nh = 10$, loss: cross-entropy + LM loss with $\alpha = 1000.0$, input embedding: 50-d pft, $bs = 64$ and $dp = 0.3$; and a GRU encoder for the ensemble with parameters, loss: LM only loss with $\alpha = 1000.0$, input embedding: 50-d pft, $nl = 4$, $kln = 10$, $bs = 64$ and $dp = 0.2$.

For augmented query, our best performing models were: (1) a BiGRU trained with parameters, loss function: LM only loss with $\alpha = 1.0$, input embedding: 50-d $skip - gram$, $nl = 3$, $kln = 5$, $bs = 128$ and $dp = 0.3$, and (2) a Transformer trained with parameters, loss function: cross-entropy + LM loss with $\alpha = 1000.0$, input embedding: 50-d $skip - gram$, $nl = 4$, $kln = 5$, $bs = 128$ and $dp = 0.3$.

While we obtain significant improvement over the compared baselines with our best-performing models, we believe further gains are possible by a more targeted search through the parameter space.