Using a Bilingual Dictionary to Expand Topic Model Training Data

Kevin Prehn and Dr. John Jeffrey

Abstract— **This project investigates artificially expanding the vocabulary of topic model training data using a bilingual dictionary. Topic classification is related to determining semantic similarity, which has wide implications for tasks like document summarization, machine translation, and information retrieval. However, topic classification models may have lower accuracy if the text used to create them does not cover a large vocabulary. Previous work used semantic graphs like WordsNet or a very large corpus to overcome this problem, but both of these methods need a lot of data and resources. Low Resource languages are language which lack sufficient amounts of these resources to take advantage of many advanced Natural Language Processing (NLP) applications, including topic classification. To work towards overcoming this problem, we investigate a different method of overcoming a limited training data using a bilingual dictionary for Low Resource Languages. We hypothesize that creating a model with expanded training data is more accurate with topic classification than a model created with non-expanded training data. After evaluating the method with a news article topic classification problem, the results determined that this method does not significantly change the performance of classification models.**

I. INTRODUCTION

Words are semantically similar if they are used in the same way, are used in similar contexts, or have similar meaning [12] Being able to determine how similar two words or pieces of text are has diverse applications, including paraphrase identification, machine translation evaluation, information retrieval (IR), text classification, text summarization, and more [3, 9]. Many of these and other tasks in Natural Languge Processing (NLP) require a large amount of training data or knowledge bases. Low Resource Languages are lacking this data or these resources and therefore can't take advantage of NLP applications like topic classification. By exploiting a bilingual dictionary, we hope to artificially expand the training data of Low Resource Languages to improve the performance of models for different Natural Langauge Processing tasks.

The goal of the testing problem for this project is to partition a set of documents according to their topics. One way that a topic can be represented is with a Probabilistic Topic Model, which is a language model that is statistically derived from analyzing "the words of the original texts to discover the themes that run through [them]" [2]. A topic model can be modeled as a unigram language model, where the probability of generating each word is independent from any other word. Despite its simplicity, this kind of language model is usually sufficient for topic classification tasks [14].

We use Probabilistic Latent Semantic Analysis (PLSA) calculate the probability distribution over a finite set of topic models. PLSA may be defined as a Corpus-based similarity measure that determines similarity between documents according to information gained from large corpora [3] The accuracy of the topic models depend on the accuracy and completeness of the training data used to calculate them. Because the system can only use the information that is available to the system [6]. If we want our topic model to be more accurate recognize a larger vocabulary, a larger training data set, or a corpora for textual data, is needed.

Rather than using a large corpus, this project investigates artificially expanding the corpus used for topic model calculation using a bilingual dictionary. This approach is inspired by semantic networks and query expansion. A semantic network is a database defining relations, word senses, and categories for terms [12]. Query expansion is used in Information Retrieval (IR), where a system adds related or more general terms to a users query to improve retrieval performance [6]. A bi- or cross-lingual dictionary can be used like a semantic network used to artificially expand a corpus to improve the topic model trained from that corpus, much like how a query is expanded to improve results in an IR system. The intuition behind this is that cross-lingual dictionaries often list many potential translations for a term. It may be assumed that these potential translations all have a semantic relationship with the original word. The hope is that this method could be used in other NLP tasks to improve the performance of models for Low Resource Languages.

II. BACKGROUND

Before comparing and analyzing two documents or bodies of text, they must be converted into a machine readable representation. One popular representation of a document is a Bag-of-Words (BOW), which is a map of words to their count in the document. In practical application, this is represented as a hash map where each word in the vocabulary is a key mapping to the probabilty of that word being generated. However, BOW is limited because it ignores word order and relies on two documents using the same terms. In [8] they overcome this by representing documents with synets instead of words. A synet is just a unique number that represents some concept; words are mapped to the synet that represents the concept of that that word. While this solves the problem of documents not being comparable without using the same terms or even the same language, it relies on the quality of the word-to-synet mapping. One benefit of this method is that it makes cross-lingual document comparison possible. Since there may not be a direct one-to-one translation between two works, it my be better to compare concepts instead of words for cross-lingual comparison [10]. Documents may also be represented by a set of N-grams, which are tokens consisting

of pairs or groups of words instead of individual words. This representation is more accurate, but it is also more expensive because the number of N-grams in a document is much greater than the number of words or size of the document.

Other than the words themselves, additional features added to the document representation can improve the quality and accuracy of the representation. A salient sentence approach calculates the most important sentences in the document based on their relationship to other sentences. The words in salient sentences are weighted more than other words. Named Entity Recognition (NER) can also improve the document representation features by labeling and categorizing words as nouns or verbs, etc. However, this relies on the accuracy of the NER system [5]. The words in the vector representation of documents are usually weighted by the term-frequency, inverse document frequency calculation (TF-IDF), which can help balance very common but insignificant words from being misjudged as too important, while making less common words more important [6].

Documents may be categorized and compared using knowledge-based methods or corpus based methods. Knowledge-based methods are semantic similarity methods which rely on a human or computer created semantic nets. The most popular one for English is WordNet. It includes the word sense for the words and categorizes all words hierarchically. Multi-lingual versions of words net also exists, but the ones in English are usually higher quality [10]. Other researchers have also exploited Wikipedia as a semantic network, since it is freely available and the largest encyclopedia to ever exist. Wikipedia encourages hyperlinks to other related pages; this is the most important property of Wikipedia that makes it usable and successful in many natural language applications. [13]. Corpus based methods "determines the similarity between words according to information gained from large corpora" [3]. In [1], they used corpora to determine the context of a concept, which is the words found surrounding an item or word representing a concept. The idea is that two words that are semantically similar will be used in similar contexts. This is also used by Googles method of modeling words called word2vec, where it predicts the surrounding words of each word (Stanford lecture). However, this method may require a very large corpus to be effective.

The topic of a text is the general main idea discussed in that text [14]. Some topics are therefore more likely to have certain words related to the particular main idea. One common method of representing topics are Probabilistic Topic Models (PMT), which are Statistical methods that analyze the words of the original texts to discover the themes that run through them. In this project, a topic model is a language model that is mapped to some topic. Badenes-Olmedo et al. determine the topic model distributions, that is, the probability distribution over possible topic models for a given document, using Latent Dirichlet Allocation (LDA). [10] evaluated topics using Explicit Semantic Analysis (ESA), using BabelNet/Wikipedia article names as the concepts, relating them according to the links between the associated Wikipedia articles. The algorithm used in this project is Probabilistic Semantic Analysis (PLSA), which also outputs a probability distribution over a fixed set of topics given a document. In PLSA, the topic model distribution can be calculated using the Expectationmaximization algorithm. The full algorithm is described in [14].

Semantic similarity is sometimes a subtask in information retrieval systems. IR systems are systems where a user provides a query and the system finds a subset of a set of documents that are relevant to the query. Semantic similarity is useful to determine if a document or a query have similar meaning. Sometimes it can be difficult for a user to find a desired document if they do not use or know the right keywords. Query expansion overcomes this by adding related or general terms to the query before searching for relevant documents [6]. In [7], they determined the similarity between a query and a document using a relevance model, which is a language model that is assumed to have generated both the document and the query, rather than just trying to find what language model generated a document.

One metric that is commonly used in IR systems is the Mean Average Precision (MAP) measure, which is a combination of two metrics: Recall and Precision. Recall is the ratio of selected documents that are relevant out of all possible selected documents. Precision is the ratio of how many selected documents are actually relevant. Recall evaluates how good a system is at retrieving all relevant information, while Precision measures how accurate those select results actually are to the query. These metrics are also used in evaluating semantic similarity, and they will be used in this project [6].

Rather than having both training data and testing data separately, researchers will use X-fold cross validation, which will train a model and evaluate it using different partitions of the same dataset [9]. However, because using a single dataset for both training and testing may mean that all of the data has come from the same source, the model evaluation may not be accurate for situations where the model works with data from a different source.

III. Method

A. Algorithm

Let D be set of all documents and let $T \subseteq D$ be the training documents. A document is represented as a BOW. V is the vocabulary of every unique word in all documents T. Each topic model in θ has a corresponding topic from a set of topics C.

The input for the algorithm is a set of input documents $I \subseteq D$ such that $I \cap T = \emptyset$, a set of unigram topic models θ, a background unigram language model b, and the probabilty that any word was generated from the background model. The output is a probability distribution over θ for each document in D. The distribution can be represented as a vector of length $|\theta|$. How this distribution is calculated is described in [14].

The topic models for concept $c_i \in C$ are generated by first calculating a normalized BOW of all training documents that

correspond to c_i, weighted using TF-IDF [6].

$\forall t \in T$, t corresponds to exactly one concept $c \in C$. Let $T_i \subseteq T$ be all training documents which correspond to the topic c_i. T_i is used to generate topic model $\theta_i \in \theta$ for c_i such that $\forall w \in \theta_i$,

$$P(w|\theta_i) = \frac{\sum_{t_i \in T_i} count(w, t_i)}{docfreq(w, T_i) \sum_{t_i \in T_i} \sum_{w' \in t_i} count(w', t_i)}$$

where $docfreq$ gives a count of how many documents in T_i the word w appears i, which serves as the TF-IDF weight adjustment.

B. Expanding Training Data

The experimental step is expanding the training documents using a bilingual dictionary. The intuition is that translating a word to one language and then retranslating it back can give a list of possible translations that are semantically similar. The hope is that this would help solve the vocabulary coverage problem by artificially increasing the diversity of words in the training data without significantly changing semantic meaning.

Let L_1 and L_2 be two different languages. Two dictionaries modeled as functions, $d12$ and $d21$ are defined such that

$$d12 : L_1 \rightarrow 2^{L_2}$$

$$d21 : L_2 \rightarrow 2^{L_1}$$

Since some words may have many possible translations, we limit the number of possible translations returned to some max value h. Expanding a training document $t \in T$ that is in the language L_1 begins with finding up to h possible translations of each word in t, creating a new document t' that is in the language L_2.

$$t' = \bigcup_{w \in t \mathcal{MS}} d12(w)$$

The $\mathcal{MS}$ means it is a multi-set union, since there may be duplicates of translated words. If there are duplicates of words, then it implies that there are multiple semantically related terms in t, and these duplicates should remain so the topic model estimates that they will be generated at a higher probability to maintain semantic meaning. t' can be used to find the final expanded document t^e in the language L_1.

$$t^e = \bigcup_{w' \in t' \mathcal{MS}} d21(w')$$

t^e will have h^2 times more words than t. Expanding all training documents in D will make D^e. θ^e is the set of topic models calculated from the expanded training documents. It is calculated the same as θ.

IV. Results

A. Experiment

To evaluate whether models using expanding training data with a bilingual dictionary are better at classifying documents than a model that does not expand training data, three different sets of topic models were created using three different sets of training data. The control training data set had only created with single articles. The expanded training data set had expanded versions of every article in the normal data set. The large corpus training data set had many documents for each topic rather than only one in the control set. The large corpus set simulates scenarios where a lot of training data is available. We wanted to see how creating topic models using expanded training data would compare to the performance of topic models created using a large amount of training data.

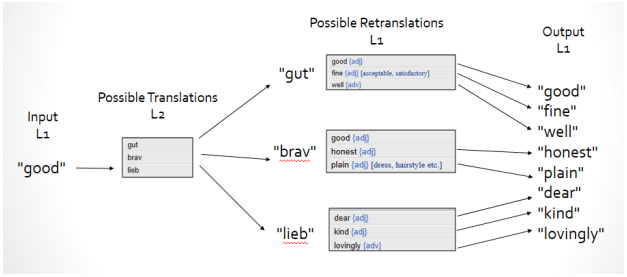

Fig. 1. Example expansion using dict.cc data to expand the world "good"

The training documents T for each topic in C are all Wikipedia articles. Following the method of [1], we assumed that a Wikipedia with a title name equal to the concept name contains only text for that topic.

The testing data is New articles collected using the New York Times Article Search API [11]. It is important that the testing data is from a different source than the training data so that is it less likely that there will be a common vocabulary, since improving vocabulary coverage is what is being evaluated.

The topics used in the experiments include: Politics, Sports, Science, and Business. Testing articles were extracted from their respective sections. The topic label for all articles is the name of the section it was collected from. Each of the four topics had three different sets of training data and therefore three topic model versions. This also makes the test more realistic because real life general applications are not guaranteed to have training and testing data from the same source.

Each topic had three different topic models for each of the three different variations of training data. The normal topic models θ were created using only one Wikipedia article for each topic, that is, the article whose name is equal to the concept or topic name. The expanded topic models θ^e were created by expanding the normal training data article with a bilingual dictionary. We used data from the English-German bilingual dictionary dict.cc [16]. The large corpus topic models θ^l were created using all Wikipedia articles that have a link in the article associated with the topic.

During topic probability distribution calculation, the background model b was a unigram language model with a uniform probability distribution over the 1000 most common words according to Google's most common words on the internet lists [15]. The probability of any word being generated by the background model was 70 percent.

Each set of topic models were used to calculate probability

Topic	θ	θ^e	θ^l
Politics	978	4,217	47,563
Sports	1,258	9,049	49,810
Science	2,356	5,509	96,730
Business	819	3,873	48,619

TABLE I

SIZE OF VOCABULARY FOR DIFFERENT TOPIC MODEL VARIATIONS FOR EACH TOPIC.

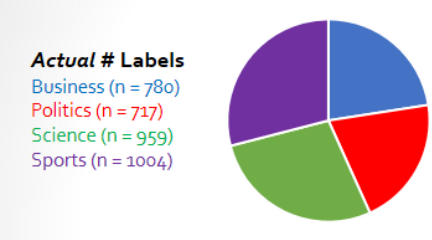

Fig. 2. Label distribution for the actual documents

distributions over topics for each article. Whichever topic model had the highest probability in the distribution was the topic that that article was classified as. Figures 2-5 shows the distribution of documents. There were 780 Business articles, 717 Politics articles, 959 Science articles, and 1004 Sports articles. All articles were collected using the New York Times API and were from the years 2017 and 2018 only. The average article word count was 929.74. The 25th percentile had 565.75 words per article, the 50th percentile had 913 words per article, and the 75th percentile has 1,203 words per article. Figure 6 shows the precision score for each variation of the model separated by topics. Precision is the percentage of correctly labeled articles out of all articles with that label. Figure 7 shows the recall score for each variation of the model separated by topics. Recall is the percentage of the labeled articles out of all of the actual articles of that label. Figure 8 shows the F1-Score of each topic model variation separated by topic. The F1 is the harmonic mean of Recall and Precision and gives an overview on the performance of a model in an information retrieval task [6], which is being modeled as a classification problem in these results.

V. DISCUSSION

The results show that the test used to evaluate the proposed method was flawed because of a high rate of incorrect classification for all models created using all training sets.

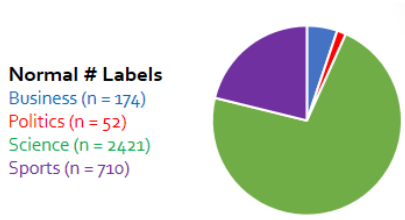

Fig. 3. Label distribution from the model created using θ

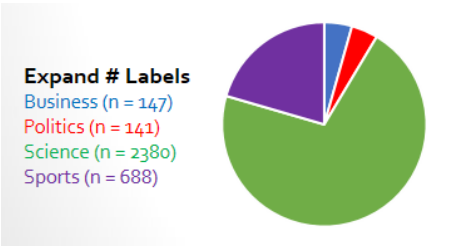

Fig. 4. Label distribution from the model created using θ^e

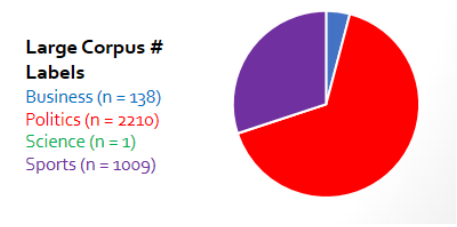

Fig. 5. Label distribution from the model created using θ^l

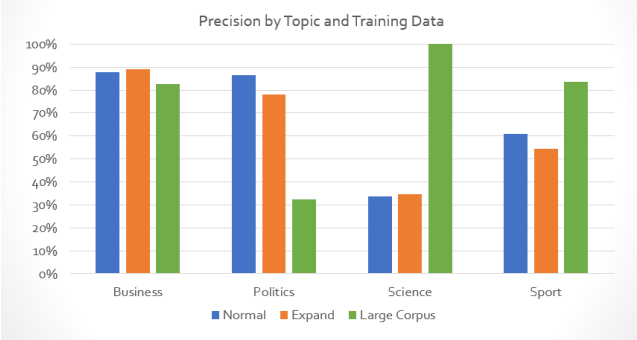

Fig. 6. Precision graph of the results separated by topic

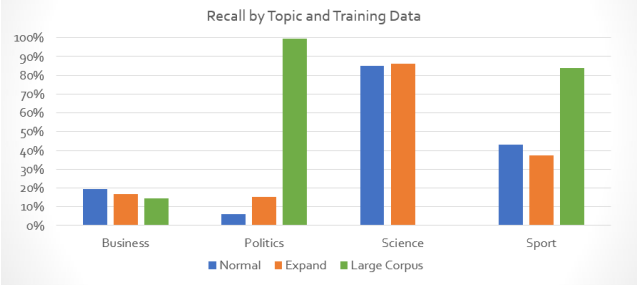

Fig. 7. Recall graph of the results separated by topic

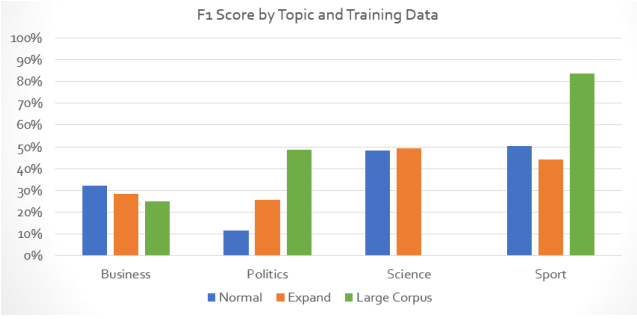

Fig. 8. F1 score graph of the results separated by topic

Comparing the distribution of labels in Figures 2-5 with the other tests, one can see that the number of Politic labels in the models generated using the large corpus training data set is far greater than any other labels, implying that this was overclassified. The same trend happens with the normal and expanded training models but with the Science label instead. Several factors could have caused this. The first could be that the training data selected was not appropriate for the classification task. While the selection of taking more generalizable training data from a different source other than the testing data was intentional, it means that the classification is expected to be less accurate. However, we argue that this is a more realistic situation that really challenges the practicality of the method. Another reason could be the choice of topics themselves; each of the news categories are very broad and could have had a lot of overlapping vocabulary, making it difficult for a model to distinguish between two different topics. The only topic that seems to not have suffered extreme misclassification was Sports based on its recall and precision. Because the vocabulary of Sports tends to be more less diverse (e.g. Basketball, Baseball, Playoffs) than the other topics, it makes sense that it suffered less from misclassification compared to the others.

Although the experiment was inherently flawed, one may still use the results to determine that expanding the training data using the proposed method does not have a significant impact on the quality of classification models. There is no clear pattern when comparing the performance of the normal and expanded models. Any difference in performance is not significant. Additionally, because both the normal and expanded models suffered from the same misclassification problems such as grossly over-classifying the Science topic, this could be a sign that both models behave similarly in the classification task overall. Comparing the performance of the expanded and large corpus models, they clearly do not have similar performance and do not suffer from the same misclassification problems. This suggests that the expanded training data model does not have significant impact on the performance of a model.

One should use caution over-generalizing the results of this test because the test itself was seemingly flawed. Although the data does not seem to support it, one should not rule out the possibility that this method would perform differently on a different classification task or with a different NLP task.

We acknowledge that there are also many problems with this method. If a low quality bilingual dictionary is used, then expanded training data may be worse than the original data. Importantly, this method ignores different word senses, or different contexts and usages of the same word, which may greatly mislead the model. Further work could use additional dictionary resources such as sample sentences to add this word-sense component.

VI. CONCLUSIONS

This project investigated a novel method for artificially increasing the amount of training data for a Low Resource Language. Many advanced NLP applications require sufficient training data, which is something Low Resource Langauges lack by definition. The results indicated that the test used to evaluate this method was inherently flawed. Despite this, the results indicate that the proposed method does not significantly alter the performance of the original training data set and that it is most likely not an appropriate replacement for a large amount of training data for classification tasks.

Future work could utilize a standard and well understood experiment or dataset to reduce the chance that misclassification was caused by overly general training data or difficult to classify testing data. Rather than include only a few topics, future work could try increasing or decreasing the number of topics as well as the nature of the topics to see if this has an impact on the performance of the different model variations. The training data could be further preprocessed before being used in a topic model such as applying word senses or part of speech tagging, but this could deviate from simulating a Low Resource Language application. The source code to the document classification, article collection, and document expansion processes used in the experiments can be found on Github at https://github.com/kvn-prhn/fall18-495-project.

ACKNOWLEDGMENT

We would like to thank the Elmhurst College Honors Program and the Department of Computer Science and Information Systems for the opportunity to conduct this research and the assistance provided.

REFERENCES

[1] A. Alba, A. Coden, A. L. Gentile, D. Gruhl, P. Ristoski, and S. Welch, Multi-lingual Concept Extraction with Linked Data and Human-in-the-Loop, Proceedings of the Knowledge Capture Conference. ACM, December, 2017.

[2] C. Badenes-Olmedo, Efficient Clustering from Distributions over Topics, XXX 2017.

[3] W. H. Gomaa and A. A. Fahmy, A Survey of Text Similarity Approaches, International Journal of Computer Applications, vol. 68, no. 13, pp. 13-18, April, 2013.

[4] G. Groe-Blting, C. Nishioka, and A. Scherp, A Comparison of Different Strategies for Automated Semantic Document Annotation, Proceedings of the 8th International Conference on Knowledge Capture. ACM, 2015.

[5] S. Kim, K. Toutanova, and H. Yu. Multilingual Named Entity Recognition using Parallel Data and Metadata from Wikipedia, Association for Computational Linguistics, July, 2012.

[6] D. H. Kraft and E. Calvin, Fuzzy Information Retrieval, Morgan & Claypool Publishers, 2017.

[7] V. Lavrenko and W. B. Croft, Relevance-Based Language Models, ACM SIGIR Forum, vol. 51, no. 2, pp. 260-267, August, 2017.

[8] P. Lops, C. Musto, F. Narducci, M. Gemmis, P. Basile and G. Semeraro, Cross-Language Personalization through a Semantic Content-Based Recommender System, Springer Artificial Intelligence: Methodology, Systems, and Applications, vol. 6304, 2010.

[9] C. Manning and R. Socher, Natural Language Processing with Deep Learning, Lecture at Stanford University, USA. 2017 [Online]. Available: https://www.youtube.com/watch?v=ASn7ExxLZws

[10] Narducci, Fedelucio, et al. "Concept-based item representations for a cross-lingual content-based recommendation process." Information Sciences 374 (2016): 15-31.

[11] New York Times Article Search API. New York Times. https://developer.nytimes.com/article_search_v2.json#/README.

[12] J. Ramapraba, S. Das, and P. Mukerjee, Survey on Sentence Similarity Evaluation using Deep Learning, Journal of Physics: Conference Series, vol. 1000, no. 1. IOP Publishing, 2018.

[13] A. Saif, N. Omar, M. J. Ab Aziz, U. Z. Zainodin and N. Salim, Semantic concept model using Wikipedia semantic features, Journal of Information Science, vol. 44, no. 4, pp. 526-551, 2018.
[14] C. Zhai, Probabilistic Latent Semantic Analysis PLSA, Lecture at University of Illinois at Urbana-Champaign, USA. 2016 [Online]. Available: https://www.youtube.com/watch?v=vtadpVDr1hM
[15] https://github.com/first20hours/google-10000-english
[16] Hemetsberger, Paul. "dict.cc Deutsch-Englisch-Wrterbuch." https://www.dict.cc. Accessed data 1. Nov, 2018.