Aligning Heterogeneous Language Vocabularies using Word-Embeddings

Simón Roca-Sotelo[1], *Jesús Cid-Sueiro*[1], *Vanessa Gómez-Verdejo*[1],
Jesus Fernandez-Bes[2], *Jerónimo Arenas-García*[1]

[1]ML4DS, DTSC, Universidad Carlos III de Madrid, Spain
[2]BSICoS Group,I3A, IIS Aragn, University of Zaragoza, Spain
[2]CIBER-BBN, Zaragoza, Spain

sroca@ing.uc3m.es

Abstract

In this paper, we deal with the problem of measuring distances between language entities, such as sentences, documents and topics derived from textual corpora. For such a task, we can rely on measures of semantic distances among words, such as those given by word-embeddings, and extrapolate them to deal with more complex entities. We show that this approach is particularly convenient when dealing with entities from heterogeneous corpora, since otherwise the presence of different vocabularies may lead to meaningless distance values. As a preliminary validity test, we show the appropriateness of our proposal by matching topics learned from NSF funded projects in different time intervals. Other use cases that could benefit from this approach, including the development of semantic distances for corpora using different languages, are discussed and will be the subject of future research.

Index Terms: Heterogeneous corpora, word embeddings, topic models, matching

1. Introduction

During the past decades many applications were created and evolved to manage digitally stored knowledge. Books, articles, webpages, opinions, services and a large etcetera are being collected and processed. Search engines, digital libraries, researchers and other agents face the problem of dealing with a crescent collection of human language and trying to extract relevant conclusions about its content.

In this paper, we deal with the problem of measuring the semantic similarity among different language entities, namely documents and topics (as defined below), paying special attention to the case in which these belong to heterogeneous corpora, in the sense that the vocabularies that characterize the corpora differ in the component words or their statistical use. This includes as a special case learning similarities between documents and topics in different languages.

Topic models characterize and cluster documents according to their implicit themes, in an unsupervised manner, assuming a latent generative model. One of the first such models was Latent Semantic Analysis [1], but the popularization of these techniques is probably due to Latent Dirichlet Allocation (LDA) model [2]. LDA assumes hidden structures between random variables modeling the word appearance as a sample of a topic distribution, and the documents as a mixture of topics.

There are many newer techniques, which sometimes differ on its latent model, whose aim is to face specific problems, or even the original algorithm is applied in unexpected contexts. Examples are found in genetics [3], image classification [4] or Information Retrieval [5]. Some variations that were proposed try to cover more complex structures not considered in the original techniques, like topic correlation [6], topic hierarchy [7], or topic time evolution [8].

Topic models provide a straightforward way to measure the semantic distance between documents by computing some probabilistic divergence between the vectors characterizing the documents in the topic model [9]. Semantic distance between topics (and not just their co-ocurrence) can also be estimated by comparing the vocabulary distributions of any two topics under the topic model. However, as far as we know there is no popular model yet which permits matching topics and documents from different corpora. This is so, because the vocabularies used for the documents in the heterogeneous corpora can differ with respect to the included words, or just because word use frequencies are corpus-dependent. In such a case, existing approaches for measuring topic and document similarities may lead to bad results.

For instance, imagine two datasets, one containing the technical proposals of projects submitted for evaluation, and a second one that characterizes the evaluators using the collection of papers authored by them. We would like to connect evaluators and proposals, but the heterogeneity between the document sources suggests that a common topic model may not a good choice. In such a case, learning separate topic models and matching the resulting topics seems more appropriate. We can think of many other similar situations, e.g., matching profiles characterizing job posts to profiles of official studies characterized by their program descriptions, or matching patents granted by offices from different countries and written in different languages.

There have been a few attempts to face similar matching scenarios. After the preliminary works of [10], [11], recent contributions are based on modified generative models that explicitly account for heterogeneity among the sources. For instance, [12] assumes a hierarchical model in which parent topics are matched to different topics for each corpus using explicit relations among words, so that any topic can be expressed differently when particularized to each corpus. The problem with this approach is that it does it assumes a perfect matching for each identified topic (parent and descendants), and the fact that, since the model is learned jointly, topics for each dataset may be less

This work has been partly supported by MINECO projects TEC2014-52289-R and TEC2016-81900-REDT, and Comunidad de Madrid project S2013/ICE-2933. Simón Roca-Sotelo has received financial support through the "la Caixa" Fellowship Grant for Doctoral Studies, 'la Caixa" Banking Foundation, Barcelona, Spain.

intuitive than if differentiated topic models were optimized. Another recent work [13] assumes that there are unobserved links between heterogeneous documents, that can be used to align the vocabularies. Identifying some of these paired documents using expert advice, a single model can be learned, obtaining topics that are characterized by words from the two vocabularies. Again, limitations of this work is that a unique topic model is learnt for both datasets, and specialized topics may be preferred in many applications. The requirement of some expert annotations can also be a limiting factor in many applications.

In this paper, we test different transformations of topics and documents that derive from word dense vector representation. This idea was firstly proposed by Bengio [14] as a way to take advantage of neural networks to learn vector representations of fixed dimensions for words. Concretely, we are using the one coming from Word Embeddings, a representation suggested by [15] which is able to capture semantic relationships between words studying their local context, instead of the global context considered in Topic Modeling techniques.

Some hybrid algorithms have been introduced for diverse applications, such as lda2vec [16] or GaussianLDA [17]. The first one learns document-level mixtures of topic vectors, combining local and global contexts as they alternate Word Embeddings and Topic Models respectively. The second one assumes that documents are sequences of Word Embeddings, and for that observations are real-valued vectors.

This paper is structured as follows: Section 2 introduces notation and definitions of the variables we want to relate through our matching models, separating those coming from the topic modeling part and the word embedding part. Section 3 specifies the different similarity distances among topics and documents from hetoregeneous corpora. Section 4 presents a first experiment to assess the appropriateness of our approach to match entities from heterogeneous corpora, providing a series of significant examples to compare the outcome of the different method. Finally, Section 5 contains the main conclusions of our work and declares the next steps in our research: proposing a formal generative model offering implicitly matching tools and establishing a formalism to compare matching approaches in a systematic way.

2. Background

In this section, we provide a very brief description of topic models and word embeddings, introducing the notation that will be necessary in subsequent sections.

2.1. Topic Models

There are several algorithms dealing with large collections of documents which learn topics from their word composition. Some examples are Latent Semantic Analysis, Latent Dirichlet Allocation or Correlated Topic Models. They differ on their assumptions, supervision degree, etc. We will only consider those cases in which we can define documents and topics as distributions of topics and words respectively.

To be more specific, we will consider that we want to characterize two different corpora, possibly using disjoint vocabularies, for which we train independent topic models using any algorithm whose output can be expressed in the following way:

- Documents topic distribution:

$$p(t_k^{(1)}|d_j^{(1)}) = \theta_{jk}^{(1)}, \tag{1}$$

$$p(t_{k'}^{(2)}|d_{j'}^{(2)}) = \theta_{j'k'}^{(2)}, \tag{2}$$

  where the superindex is used to denote the corpus, $k = 1, \dots, T^{(1)}$ and $k' = 1, \dots, T^{(2)}$ index the topics of the first and second topic models, $T^{(1)}$ and $T^{(2)}$ being the number of topics in each model, whereas j and j' similarly index the documents of each corpus.

  For the probabilistic topic models we use, the topic-document proportions constitute a consistent probability distribution, i.e., $\theta_{jk}^{(1)}, \theta_{j'k'}^{(2)} \geq 0$,

$$\sum_k \theta_{jk}^{(1)} = 1 \text{ and } \sum_{k'} \theta_{j'k'}^{(2)} = 1,$$

  for any documents $d_j^{(1)}$ and $d_{j'}^{(2)}$.

- Topics are characterized as distributions over words:

$$p(w_l^{(1)}|t_k^{(1)}) = \beta_{lk}^{(1)}, \tag{3}$$

$$p(w_{l'}^{(2)}|t_{k'}^{(2)}) = \beta_{l'k'}^{(2)}. \tag{4}$$

  Here, l and l' index the words in the two vocabularies. As before, we have that $\beta_{lk}^{(1)}, \beta_{l'k'}^{(2)} \geq 0$,

$$\sum_l \beta_{lk}^{(1)} = 1 \text{ and } \sum_{l'} \beta_{l'k'}^{(2)} = 1,$$

  for any topics $t_k^{(1)}$ and $t_{k'}^{(2)}$.

  For convenience of notation, we define vectors $\boldsymbol{\beta}_k^{(1)}$ and $\boldsymbol{\beta}_{k'}^{(2)}$ as column vectors containing probabilities $\beta_{lk}^{(1)}$ and $\beta_{l'k'}^{(2)}$, respectively. Note that, since we are not assuming the same vocabulary for both corpora, these topic-defining vectors will in general be of different length for the topics of the first and second models.

2.2. Word Embeddings

Word Embeddings as in [15] provide dense vector representation for words coming from huge vocabularies, for instance, Wikipedia articles. This representation, which is connected with the recent research in Deep Learning algorithms, captures semantic information on local word context.

In a simple illustrative conception, words are represented as BoW in the input. Then, one hidden layer is set with as many neurons as dimensions we want for our vector. The output has the same dimension as the input, but representing the probability of each word to appear in the context of the input word.

Finally, setting a proper cost function, this problem converges in a way that weights of the network can be seen as a look-up table, in which the ith row corresponds to the vector representation of the ith word. One intuition for similarity could be that semantically close words should appear in similar contexts, and for that the output should be related. This is obtained through similar weights.

This approach has proven its utility to find geometric similarities between words, and the typical similarity measurement is the cosine similarity between their vectors. In this paper, we denote the vector representation of words using bars, e.g., $\bar{\boldsymbol{w}}_l^{(1)}$, and $\bar{\boldsymbol{w}}_{l'}^{(2)}$, whereas the cosine similarity between two words in embedding space will be denoted as $W(\bar{\boldsymbol{w}}_a, \bar{\boldsymbol{w}}_b)$.

3. Similarity among Topics and Documents from heterogeneous Corpora

In this Section, we present different strategies for computing similarities among topics and models from heterogeneous corpora.

Two models are suggested in this contribution, one of them based on Word Embeddings. In the first model, before the matching starts, we need to preprocess both corpuses in order to have a common framework which considers both vocabularies. This implies:

1. Designing a common vocabulary.
2. Create a Bag of Words (BoW) describing all documents based on the same dictionary.
3. Assuming all topics as a disjoint union, which implies that distributions may need to be normalized.

It is not our purpose to re-launch our topic model algorithm with a combined corpus. Instead, we look to define similarity measurements based on parameters learned during the two separated training processes.

The first step is intended to construct a common vocabulary representation for both corpora, while reducing the overall size vocabulary. This is important not just for a more efficient computation, but also to remove many terms which are not particularly defining any topic, so that keeping them would just result in added noise for the similarity estimation function.

In order to create a common vocabulary, we select the most relevant words of each topic of both topic models. In order to do so, we could select a fixed number of words per topic, or select as many words as necessary so that their joint probability (for that particular topic) exceeds a threshold value. More details on this will be given in the experiments section.

Once the common vocabulary is constructed, we use it to map the Bag of Word (BoW) representation of documents from both corpora. Under this representation, each document features consist of a word-count vector. This BoW representation will be denoted as $\widetilde{\mathbf{B}}_j^{(1)}$ and $\widetilde{\mathbf{B}}_{j'}^{(2)}$ for the documents of the first and second corpora, respectively.

Similarly, we map the topic vectors on this common representation space. We denote as $\widetilde{\boldsymbol{\beta}}_k^{(1)}$ and $\widetilde{\boldsymbol{\beta}}_{k'}^{(2)}$ the topic distributions over the common vocabulary, which requires also normalization after some words have been discarded.

3.1. Strategy 1: High co-occurrence of words

In this first approach no Word Embeddings are used. Basic intuitions will be followed when setting similarity measurements. For instance, two topics are similar when they use the same words in a similar proportion. Although probability divergences may be used for this, we have found that inner products among the vectors provide qualitatively similar results, while being much faster to compute.

$$S(t_k^{(1)}, t_{k'}^{(2)}) = \widetilde{\boldsymbol{\beta}}_k^{(1)T} \cdot \widetilde{\boldsymbol{\beta}}_{k'}^{(2)} \tag{5}$$

$$S(t_k^{(1)}, d_{j'}^{(2)}) = \widetilde{\boldsymbol{\beta}}_k^{(1)T} \cdot \widetilde{\mathbf{B}}_{j'}^{(2)} \tag{6}$$

$$S(d_j^{(1)}, d_{j'}^{(2)}) = \widetilde{\mathbf{B}}_j^{(1)T} \cdot \widetilde{\mathbf{B}}_{j'}^{(2)} \tag{7}$$

These are by far no formal similarity/distance metrics, since they do not follow all of the properties of non-negativity, identity of indiscernibles, symmetry and triangle inequality. However, they provide a first approach for solving the matching, since this measurement should be higher in those cases in which variables are non-zero – e.g., same word appearing in two different topics –, and also when probabilities are high – high word occurence following previous example. Nevertheless, this measurement makes no consideration at all with exclusive words from each corpus–.

For sure, it can be argued the interest of this approach. Equal or "similar" vocabularies may allow the success of the matching process. However, very heterogeneous vocabularies will make difficult to match topics or documents when they are defined by different words.

3.2. Strategy 2: Additive Compositionality Based on Word Embeddings

Secondly, we consider a model in which topics and documents from different corpuses are represented first in the same vector space, without constructing an auxiliary common vocabulary. This allows us to apply any vector metric to measure similarity between documents and topics.

Inspired by [16] and [18], we propose to represent topics as an average of word embedding vectors for the terms composing the topic, each of them multiplied by its corresponding probability parameter, i.e.:

$$\bar{\mathbf{t}}_k^{(1)} = \frac{1}{\sum_l \beta_{lk}^{(1)}} \sum_l \beta_{lk}^{(1)} \bar{\boldsymbol{w}}_l^{(1)}, \tag{8}$$

$$\bar{\mathbf{t}}_{k'}^{(2)} = \frac{1}{\sum_{l'} \beta_{l'k'}^{(2)}} \sum_{l'} \beta_{l'k'}^{(2)} \bar{\boldsymbol{w}}_{l'}^{(2)}, \tag{9}$$

where $\bar{\mathbf{t}}_k^{(1)}$ and $\bar{\mathbf{t}}_{k'}^{(2)}$ are the vector representations of the topics of both models in the common embedding space.

With respect to documents, we can consider two different representations, depending on whether we decide to average over the words or the topics representing a document. In the first case and following the example of topic vectors, a document vector could be constructed as the average of words composing it, considering its occurrences as frequencies summing up to one. In the second case, the topic embedding vectors should be averaged using the topic proportions ($\theta_{jk}^{(1)}$ or $(\theta_{j'k'}^{(2)})$ as weights.

Note that under this approach the embedding space provides a common representation space for words, documents and topics, so that similarities of any two entities of any corpora can now be straightforwardly computed using the cosine similarity of their corresponding representation vectors. Cosine similarity is bounded between -1 and 1, with larger values indicating higher similarity.

4. Experiments

In this set of preliminary experiments we make use of Python utilities like Gensim, NLTK or NumPy to qualitatively compare the matching between topics learned in National Science Foundation (NSF) dataset [19], which contains title, abstract and some metadata of project proposals. Two different corpora were created according to the following date intervals: "1990 to 1995" and "2010 to 2016". There are specific topic models that consider the time evolution of topics. However, here we create independent topic models for each subset of proposals. The different interval dates have been picked up in purpose, since the main advances in Science are translated into some variations and new introduced terms, that make vocabularies different but with a certain amount of coincidences. Therefore, our goal here

Table 1: Corpora and Dictionary sizes

	# Documents	Vocab. size
Corpus 1 (1990-1995)	51 451	25 836
Corpus 2 (2010-2016)	73 165	44 761

is to study whether the proposed similarity criteria allow us to match topics from the different corpora that, in spite of being characterized by different terms and/or word probabilities, are semantically close.

In the following we will refer to these corpora as Corpus 1 and Corpus 2. The number of documents and vocabulary size of each corpus is indicated in Table 1. The number of unique words that conform each dictionary is the result of a pre-processing step in which lemmatization was done, as well as word filtering in those cases which could degrade the Topic Modeling –very frequent/unfrequent words, numbers, symbols, etc–.

An LDA model has been trained for each corpus exploring the number of topics in the range from 10 to 100. It is important to pick a proper value for the number of topics, since too few topics typically result in general uninformative topics, and a too large number may lead to extremely specific topics, but also include redundant ones, or even topics that focus on the generation of unimportant but frequent words [9]. Visually inspecting the models, we decided to use 40 topics, since these models looked qualitatively adequate for our purposes. In any case, note that the focus of our work is not on the design or optimization of topic models, but on comparing topics from heterogeneous models, no matter how these are obtained.

In those points where matching relies on word embeddings, we use the english pre-trained Wikipedia embeddings coming from Facebook Research Group [20].

One important issue, possibly a bottleneck depending on the available computing resources, is the number of chosen words when considering full dictionaries is not an option. In this respect, we have explored how many unique words we need to pick from each corpus in order to retain a predefined probability threshold on all topics. Figure 1 shows the result of such analysis. This figure contains an implicit message: each corpus has a lot in common with the other one, which makes sense, but there are also a non-negligible amount of words which are important to describe the topics of just one corpus, suggesting that methods for matching topics should be aware of such different vocabularies.

Finally, two matching experiments were done, testing similarity strategies 1 and 2 to link topics between Corpus 1 and Corpus 2, due to the exploratory nature of this article. Both strategies used as many words as needed to retain 80% of probability in every topic, which resulted in a significant vocabulary size reduction (see Fig. 1).

Comparison of the different strategies is difficult, since we do not have a ground truth to compare with. Therefore, we will rely on the identification of some significant examples which illustrate well the performance of the matching strategies. In order to do so, Tables 2 and 3 show, for one topic of each corpus, the top three matched topics using strategies 1 and 2, as well as the corresponding estimated similarities. It is important to realize that measurements from strategy 1 cannot be directly compared in magnitude with those coming from strategy 2. The first one is a dot product bounded by the product of vector norms, and the second one is a cosine similarity whose

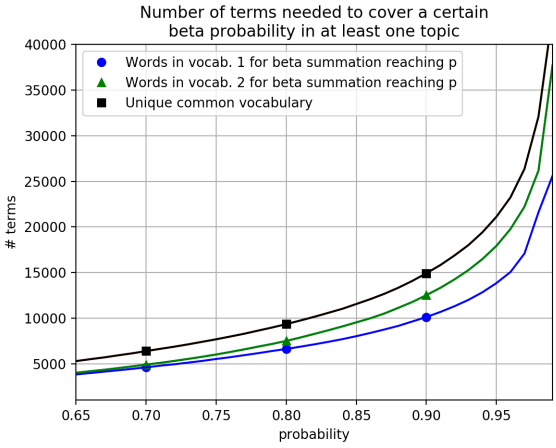

Figure 1: Number of terms needed to accumulate a certain probability in the topic model. For each probability threshold value, the most relevant terms from each topic were selected so that the accumulated probability is above the threshold. Vocabulary size is then computed as the number of different terms over the topics of the first and/or second models.

maximum value is 1. Some details about how each model operates arise, and it can be seen that:

- The lack of a formal metric description in strategy 1 leads to a measurement unable to penalize or encourage bad matches from good matches.
- Since in strategy 1 only exact word coocurrences increase the similarity score, topics with frequent terms but no very specific thematic orientation (e.g., topic $t_{24}^{(2)}$ in Table 2 or all topics selected by strategy 1 in Table 3) can show larger similarity to the target topic than other topics which are semantically more aligned.
- On the other part, strategy 1 has not been distracted by those words. It takes advantage of shared vocabulary but is also able to reward those words that belong to a similar context, as cosine similarity for Word Embeddings does. This can be appreciated for instance in the second and third topics suggested by this model, in which the scope of the topic may be different but there is still a clear link between the words of both sides.

Overall, we can appreciate that strategy 1 seems more coherent for a human spectator when scoring the similarity.

5. Conclusions and Further work

In this contribution, we have combined in an experimental way the practical benefits of Word Embeddings jointly with the implicit global context modeling of Topic Models to set the basis to a future research line. We have shown that these two techniques may bring together more knowledge about matching language entities than by themselves. The combination of the learned parameters of a generative model plus word vector representation sums details which go beyond of identifying word coincidences. Besides, recent articles from skilled researchers collect interest about new variations of these methods.

In spite of having already obtained interesting results, there are many directions to extend this work. First, we need to fur-

Table 2: Matching example for topic 7 in vocabulary 1

	Topics selected with strategy 1	Similarity estimated by strategy 1	Similarity estimated by strategy 2
Topic $t_7^{(1)}$ chemistry, reaction, chemical, molecular, molecule, metal, organic, program, structure, complex	$t_{27}^{(2)}$ robot, control, game, object, video, robotic, robotics, task, cybersecurity, autonomous	0.014	0.73
	$t_{24}^{(2)}$ nsf, scientist, workshop, international, collaboration, scientific, support, national, US, university	0.0125	0.72
	$t_{27}^{(2)}$ problem, equation, mathematical, solution, method, nonlinear, differential, numerical, application, partial	0.0122	0.80
	Topics selected with strategy 2	**Similarity estimated by strategy 1**	**Similarity estimated by strategy 2**
	$t_1^{(2)}$ chemistry, reaction, metal, catalyst, synthesis, organic, professor, process, fuel	0.011	0.98
	$t_{26}^{(2)}$ quantum, physic, state, electron, material, magnetic, theoretical, property, phase, interaction	0.01	0.93
	$t_{22}^{(2)}$ cell, protein, molecular, biological, molecule, biology, division, structure, cellular, function	0.011	0.92

Table 3: Matching example for topic 25 in vocabulary 2

	Topics selected with strategy 1	Similarity estimated by strategy 1	Similarity estimated by strategy 2
Topic $t_{25}^{(2)}$ change, ecosystem, soil, environmental, climate, forest, management, response, land, community	$t_{35}^{(1)}$ young, investigator, award, nsf, presidential, science, dr, polymer, enabling, objective	0.019	0.77
	$t_4^{(1)}$ support, program, university, award, fellowship, postdoctoral, month, science, graduate, center	0.017	0.76
	$t_{19}^{(1)}$ site, university, state, reu, experience, chemistry, ten, san, undergraduate, french	0.016	0.77
	Topics selected with strategy 2	**Similarity estimated by strategy 1**	**Similarity estimated by strategy 2**
	$t_1^{(1)}$ plant, forest, ecosystem, soil, growth, effect, community, food, insect, environmental	0.011	0.94
	$t_{33}^{(1)}$ change, ocean, climate, global, water, data, flux, model, lake, scale	0.011	0.94
	$t_{23}^{(1)}$ model, behavior, effect, social, understanding, individual, change, theory, decision, test	0.007	0.9

ther study these and other similarity measurements and its properties, trying to identify which representation features codify the most relevant information with the purpose of constructing useful formal measurements. In order to compare different similarity measures, an important step would be to develop objective measurements that align with human perception, for which we plan to carry out collective annotations. Another issue that deserves our attention is that of analyzing the tradeoff between performance and scalability, which is inherent to the introduction of mechanisms for pruning the vocabularies.

In the long term, we plan to develop methods that incorporate the matching as part of a generative model prepared to deal with heterogeneous vocabularies automatically, considering a hidden structure built on a common word space. Hopefully, such model would be able to model relationships on how two heterogeneous but related vocabularies are, indeed, connected.

6. References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[3] B. Liu, L. Liu, A. Tsykin, G. J. Goodall, J. E. Green, M. Zhu, C. H. Kim, and J. Li, "Identifying functional mirna–mrna regulatory modules with correspondence latent dirichlet allocation," *Bioinformatics*, vol. 26, no. 24, pp. 3105–3111, 2010.

[4] W. Chong, D. Blei, and F.-F. Li, "Simultaneous image classification and annotation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1903–1910.

[5] X. Yi and J. Allan, "A comparative study of utilizing topic models for information retrieval." in *ECIR*, vol. 9. Springer, 2009, pp. 29–41.

[6] J. D. Lafferty and D. M. Blei, "Correlated topic models," in *Advances in neural information processing systems*, 2006, pp. 147–154.

[7] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Advances in neural information processing systems*, 2005, pp. 1385–1392.

[8] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. Intl. Conf. Machine Learning (ICML)*, 2006, pp. 113–120.

[9] ——, "Topic models," in *Text Mining: Classification, Clustering, and Applications*, S. A. N and S. M, Eds., 2009, pp. 71–93.

[10] M. Paul and R. Girju, "Cross-cultural analysis of blogs and forums with mixed-collection topic models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 2009, pp. 1408–1417.

[11] C. Zhai, A. Velivelli, and B. Yu, "A cross-collection mixture model for comparative text mining," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 743–748.

[12] C. Chen, W. Buntine, N. Ding, L. Xie, and L. Du, "Differential topic models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 2, pp. 230–242, 2015.

[13] Y. Yang, Y. Sun, J. Tang, B. Ma, and J. Li, "Entity matching across heterogeneous sources," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2015, pp. 1395–1404.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[16] C. E. Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," *arXiv preprint arXiv:1605.02019*, 2016.

[17] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings." in *ACL (1)*, 2015, pp. 795–804.

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[19] N. S. Foundation, "Historical Awards," https://www.nsf.gov/awardsearch/download.jsp, [Online; accessed 1-July-2017].

[20] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.