Human-In-The-Loop Topic Discovery with Embedded Text Representations

Eric Bunch
Qian You
Glenn Fung
ebunch@amfam.com
qyou@amfam.com
gfung@amfam.com

ABSTRACT

Most automatic topic discovery algorithms struggle with providing topics that are interpretable for humans familiar with the corpus domain. In this paper we propose a human-in-the-loop approach where the topic discovery algorithms can be guided by a domain expert to produce topics that are meaningful to the users. The produced topics can provide meaningful and potentially actionable insights which is often the goal when using a topic discovery algorithm. We propose a set of actions that can be provided by the human at any iteration of a state-of-the-art, word-embedding-based topic discovery algorithm, WELDA. The nature of WELDA being that it alternates between fitting a standard LDA model and replacing words in the corpus leveraging an embedded word space lends itself to the interjective nature of human-in-the-loop versions of topic models (HL-TM). We call this combination of human-in-the loop with the WELDA topic model HL-TM WELDA.

We demonstrate the efficacy of our proposed approach in both publicly available data (20 Newsgroups) and industry-related data extracted from the claims workflow of a large insurance company.

CCS CONCEPTS

• **Human-centered computing → User interface programming**.

KEYWORDS

topic models, word embeddings, human in the loop

ACM Reference Format:
Eric Bunch, Qian You, and Glenn Fung. 2020. Human-In-The-Loop Topic Discovery with Embedded Text Representations. In *Proceedings of (DaSH@KDD).* ACM, New York, NY, USA, 7 pages.

1 MOTIVATION

Many industry-related processes generate vast amounts of information. Unstructured text is one of the more common and ubiquitous kinds of information being constantly generated, specially for processes or workflows that involve interactions between the company and their customers. Hence the need for efficient ways to manage, categorize and present the underlying valuable information encoded in this ever-growing steams of unstructured text.

For example, in the insurance industry, thousands of claims are reported every day. Being able to process claim's descriptions to 1) automatically categorize them and 2) discover new claim causes that could suddenly trend is paramount to be able to assign the appropriate resources so claims are handled efficiently and consequently to improve customer experience.

Claims descriptions and causes can be expressed in many different ways depending on the customer and the representative that are reporting the claim, discovering and grouping claim categories or causes (topics) can be a daunting task for a human. Therefore, there is a growing need for computer assisted topic discovery.

However, most automatic topic discovery algorithms struggle with providing topics that make quick sense to a human with domain knowledge expertise. In this paper we propose a human-in-the-loop approach where the topic discovery algorithms can be guided by a domain expert to produce topics that are meaningful to the expert and can provide actionable insights about the creation of new intents or topics of interests to customers.

Next, we explore state-of-the-art techniques for topic discovery and in the subsequent section we propose and describe modifications that allow humans to interact efficiently with such algorithms. To end the paper we demonstrate the efficacy of our proposed approach with both publicly available data and real data extracted from an insurance claim workflow.

The main contributions of this paper can be summarized as follows

- We have combined the human-in-the-loop LDA framework with the word embedding-based topic modeling algorithm WELDA to obtain HL-TM WELDA (Human in the loop topic modeling WELDA). As a far as we know this is the first human-in-the-loop topic modeling algorithm that incorporates word vector embeddings.
- We have augmented the refinement options given in [30] based on user experience and feedback.
- We have open sourced [1] the code for the algorithm as well as a web app for the user interface for the human-in-the loop portion of this work. It should be noted that neither the code for [30] nor [6] has been open sourced.

DaSH@KDD, August 24, 2020, Virtual Conference

[1] https://github.com/AmFamMLTeam/hltm_welda

2 RELATED WORK

Latent Dircichlet Allocation (LDA) [4] is a generative probabilistic topic model that generates topics by learning a topic distribution for each document and a word distribution for each topic from a collection of documents. Since LDA requires no prior knowledge of either the content of the topic nor any manual annotations, it has been a powerful tool for knowledge discovery for unstructured text.

On the other hand, several vector space word embedding models, such as Word2Vec [22], Doc2Vec [18] and Glove [26] have become popular representations for unstructured text via unsupervised learning. Word embedding models usually learn word representations based on a small, local context window around that word. Those models map words into a semantic space where word similarities are preserved regardless of misspelling, synonyms or not appearing in the same set of documents.

Therefore there have been increased efforts [2, 6, 9, 15, 24] in incorporating a pre-trained word vector model into an LDA topic model. For example, the Word Embedding LDA (WELDA)[6] proposes a simplified and elegant process of integrating a multivariate Gaussian topic distribution in a word embedding space and a Dirichlet document generation process. However most of those approach is still primarily data-driven and unsupervised. Human-in-the-loop topic modeling (HL-TM) algorithms [29] are proposed to translate human interactions to alter the underlying LDA algorithm. But there is little research of human-in-the-loop topic modeling based on a hybrid approach using both LDA and word embeddings. Therefore we have modified WELDA with a human-in-the-loop approach (HL-TM WELDA) to take advantage of the benefits word embedding representations brings. Also in contrast with [29], we have redefined / enhanced several user interactions with documents driven by user experience.

3 WELDA FOR TOPIC DISCOVERY

The effectiveness of the WELDA algorithm for discovering topics within a corpus was explored in [6], and it was also compared to the approaches in [2, 9, 15, 24]. In particular, a variant of topic coherence, denoted $_+$ in [27], was used to evaluate the quality of the topic models. It was demonstrated that WELDA almost uniformly obtained higher topic coherence than the other compared approaches.

3.1 Generative Process

WELDA enhances the traditional LDA topic model algorithm by incorporating information from pre-computed word embeddings to estimate the posterior distributions. This influences the words associated with a topic to also be coherent in the chosen embedding space.

The WELDA topic model assumes that the corpus is generated from the process summarized below, where notation is as appears in Table 1.

The generative process described in Algorithm 1 and it's corresponding notation is slightly generalized than that presented in [6] to include a matrix representation of both the document-topic priors $U_{3\bullet:}$ and the topic-word priors $V_{:\bullet F}$. This generalization is necessary for the additional functionality pertaining to the human-in-the-loop aspect of this project detailed in the sequel.

Algorithm 1 WELDA generative process

for $:$ = 1 to **do**
  Choose word distribution $:$ ∼ 8A82 ;4C($V_{:\bullet}$)
  Fit normal distribution $:$ to $:$
end for
for each document 3 in the corpus **do**
  Choose topic distribution 3 ∼ 8A82 ;4C($U_{3\bullet}$)
  for each word index from 1 to $,_3$ **do**
    Draw topic I_{3*} ∼ "D;C8=><8O;(3)
    Draw word F_{3*} ∼ "D;C8=><8O;(I_{3*})
    Toss a coin $3*$ ∼ 4A=>D;;8(_)
    if $3*$ = 1 **then**
      Replace word F_{3*} by word $F^*_{3*} \sim \mathcal{N}(`_{:\bullet}\ :)$
    end if
  end for
end for

Symbol	Description
	Number of topics
	Number of documents
$,_3$	Number of words in document 3
3	Topic distribution for document 3 ∼ 8A82 ;4C($U_{3\bullet}$)
$:$	Word distribution for topic $:$ ∼ 8A82 ;4C($V_{:\bullet}$)
$:$	Embedded topic distribution for topic $:$ ∼ $\mathcal{N}(`_{:\bullet}\ :)$
$3*$	Coin toss ∼ 4A=>D;;8(_)
F_{3*}	Word in document 3 at position 8 ∼ "D;C8=><8O;(I_{3*})
F^*_{3*}	Word corresponding to vector in embedding space to replace $F_{3*} \sim \mathcal{N}(`_{:\bullet}\ :)$
I_{3*}	Topic assignment for word F_{3*} ∼ "D;C8=><8O;(3)
	Hyperparameters
$U_{3\bullet:}$	Matrix of Dirichlet priors for document-topic distribution
$V_{:\bullet F}$	Matrix of Dirichlet priors for topic-word distribution
$\_$	Resample probability
$`_{:\bullet}\ :$	Mean and variance for topic $:$ in embedding space

Table 1: Notation

Having a topic model that uses learned word embeddings is desirable in a domain-specific setting as in insurance, where we can use a word embedding model pre-trained on a large corpus of insurance-specific text. Then WELDA can be used to discover topics on a smaller corpus (e.g. customer-agent chat data), leveraging the large corpus through injecting word vectors into the topic model. In this section, the sequel, as well as in the implementation, we use the projection of the word vectors onto the first two principal components instead of the full word vectors themselves. The reason for this is to reduce computational complexity of the WELDA model, as explored in [6], Section 4.5.

3.2 Inference

When fitting the WELDA model, we use a modification to the standard sampling equation from LDA, the collapsed Gibbs sampler [11]:

$$\%(I_{3\bullet 8} = : \mid F_{3\bullet 8} = <\bullet I_{\neg 3\bullet 8}\, o) \propto \frac{,^{)}_{<\bullet:} + V_{<\bullet:}}{,^{)}_{:<0} + V_{:\bullet<0}} \cdot \left(^{)}_{3\bullet:} + U_{3\bullet:}\right) o$$

where $I_{3\bullet 8} = :$ represents the assignment of the 8^{C} word in document 3 to topic $:$, $F_{3\bullet 8} = <$ represents that the 8^{C} word in document 3 is the $<^{C}$ word in the lexicon, and $I_{\neg 3\bullet 8}$ are the topic assignments of all other tokens. Furthermore, $,^{)}_{<\bullet:}$ is the number of times word $<$ is assigned to topic $:$, not including the current instance, and $^{)}_{3\bullet:}$ is the number of times topic $:$ has occurred in document 3, not including the current instance.

After an initializing the model with the standard LDA algorithm, we estimate Gaussian distributions in the embedding space by computing the mean $\grave{}_{:}$ and covariance matrix $\Lambda_{:}$ of each topic $:$ by using the $\#^{C>?}_{:}$ word vectors corresponding to the top $\#^{C>?}_{:}$ words in topic $:$. Then random words in the corpus are chosen using a Bernoulli distribution. If a word $F_{3\bullet 8}$ is chosen with topic assignment $:$, then a random vector $v_{3\bullet 8}$ is drawn from the Gaussian distribution $\Lambda_{:} \sim N(\grave{}_{:}, \Lambda_{:})$. Then the closest word to $v_{3\bullet 8}$ in the corpus vocabulary is chosen using a nearest neighbor search through a kd-tree, $F_{3\bullet 8}$ and the word $F_{3\bullet 8}$ is replaced in the corpus with $F_{3\bullet 8}$. After that, additional iterations of the Gibbs sampler are run.

4 HUMAN-IN-THE-LOOP TOPIC DISCOVERY

In contrast with static topic models that estimate topics based on static initial user-defined parameters, Human-in-the-loop topic models (HL-TM) provide mechanisms that allow users to refine and change topic models dynamically. There have been multiple implementations of topic models that allow feedback from a human in the loop [8, 12, 13, 19]. These implementations typically included feedback mechanisms that lent themselves to the choice of model, rather than prioritizing what users wanted in a topic discovery tool. [20] proposed a much more user-centric approach to create a topic model interface; however, the model feedback was not incorporated. The work of [30] has taken this further and has implemented a human-in-the-loop LDA model with seven refinement operations: add word, remove word, change word order, remove document, merge topic, split topic, and add to stop words. These refinements were carefully chosen based on the desire of end users, and implemented to adhere to user expectations of predictability and control.

The implementation of LDA in [30] used the collapsed Gibbs sampler, and the above refinements affect the Dirichlet priors, causing the sampler to adhere to the preference of the user. The choice of using the collapsed Gibbs sampling method to fit the LDA model is key in having an algorithm that can be modified by the user. During collapsed Gibbs sampling, an internal state of the model can be kept in-memory and directly modified in ways that immediately reflect the actions enumerated in [30]. In contrast, other methods of fitting the LDA model either do not keep an internal model state, and directly estimate the posterior distributions, or the internal model state is not easily modified in ways that reflect desirable user interactions.

However, a key limitation of the work presented in [30] is the use of the standard bag-of-words representation utilized in LDA-type topic discovering algorithms which excludes the potential that vectorial word embedding representation has to offer: lower dimensional dense representations, misspelling handling, synonyms, potentially contextual information depending on the embedding method used, and in general injecting more co-relational information than may be present in the corpus at hand.

To address this limitation, it is natural to explore adding human in the loop to a hybrid LDA model. The nature of WELDA [6] being that it alternates between fitting a standard LDA model and replacing words in the corpus leveraging the embedded word space lends itself to the interjective nature of human-in-the-loop versions of topic models. Yet unique challenges exist for HL-TM systems: on one hand those systems need to provide user experience with high predictability, control and accuracy; on the other hand, users interaction needs to be seamlessly fed back to update the underlying topic modeling process.

4.1 Refinement

As is explained in [30], we divide the user feedback into two broad classes: forgetting an error the model made, and injecting new knowledge into the model. Forgetting certain things learned by the model translates to invalidating the topic-word assignments for certain word types. Injecting information into the model translates to changing the Dirichlet prior parameter matrices $U_{3\bullet:}$ and $V_{:\bullet F}$.

The implementation of refinements in HL-TM WELDA are enumerated below. They differ from those in [30] in that the change word order refinement is dropped, and the refinements remove document from corpus and remove document from topic are added. These modifications to the refinements came after an initial round of user testing in which feedback on the refinements options was gathered. The topic refinements are:

(1) Add word to topic: to add word F to topic $:$, we forget the topic assignments for every occurrence of F. Then we set $V_{:\bullet F} = \max(^{1}_{\neg:} \cdot \,^{)}_{:\bullet:}) \cdot \,^{)}_{F\bullet:}$, and every occurrence of F is assigned a topic drawn from $"(D; C8 => <80, V_{\bullet F} o)$. This is appropriate if a word occurring in one topic seems a better fit for another.
(2) Remove word from topic: to remove word F from topic $:$, we forget the topic assignments for every occurrence of F. Then we set $V_{:\bullet F}$ to a very small value n, and the topic assignments for F are drawn from $"(D; C8 => <80, V_{\bullet F} o)$. This can be used if a word seems obviously out of place in its current topic.
(3) Add document to topic: to add document 3 to topic $:$, we forget the topic assignments for all words in 3. Then we set $U_{3\bullet:} = \max(^{1}_{\neg:} \cdot \,^{)}_{3\bullet:})$, and the topic assignments for all words in 3 are drawn from $"(D; C8 => <80, U_{3\bullet} o)$. This is appropriate if it is clear that a document is aligns more closely with a topic other than its current topic.
(4) Remove document from topic: to remove a document 3 from a topic $:$, we forget the topic assignments for all words

in 3. Then we set $U_{3,:}$ to a very small value, and the topic assignments for all words in 3 are drawn from $\mathrm{Dir}(U_{3,:}\theta)$. This can be used if a document seems obviously out of place in its current topic.

(5) Merge topics to merge topic $:_1$ with topic $:_2$, we assign $:_1$ to all words previously assigned to $:_2$. The new $U_{:_1}$ is taken to be the average of the two vectors $U_{:_1}$ and $U_{:_2}$. This can be used if the user thinks two topics are related enough to be viewed as one.

(6) Split topic to split a topic, the user specifies a subset of the topic's words (called seed words), which will be assigned a new topic $:_=$. Once the seed words are assigned topic $:_=$, we set $V_{:_=,F}$ to 1 if F is a seed word, and to a standard initializing value if F is not a seed word. This is appropriate if the user thinks that a topic is really made up of two separate topics.

(7) Add to stop words to add a word F to the set of stop words, every occurrence of F is removed from the corpus. This can be used if a word is viewed as extremely common or non-informative in the context or domain.

(8) Remove document from corpus to remove a document 3 from the corpus, the document is simply removed, as well as each instance of the words occurring in 3. This can be used to remove a document that might be considered noise or not informative.

In addition to the above refinements to the underlying model, the HL-TM WELDA tool allows users to evolve the model, which equates to iterating the Gibbs sampler, changing the name of a topic, and saving and loading the model state. Figure 1 depicts the work flow of this human-in-the-loop topic discovery process.

4.2 Interface

Similar to [30], we have developed a user interface (Fig 2 - Fig 4) represents a topic model as a list of topics on a left panel, each displayed with the topic name, the number of documents for which that topic is the most likely, and the top three words for the topic. Selecting a topic in the list in the left panel displays a more detailed view of the topic in the right panel, including the top 30 words for the topic, and snippets of the top 10 documents for the topic, ordered by the probability of topic given document 3; that is, by $\theta_{3,:} \mid 3^{\alpha}$. The user can click the document snippet and expand the row to see the entire document. The top 30 words are ordered by its probability for the topic: ; that is, by $\Phi_{F \mid :} \cdot^{\beta}$.

To access the functionality in the refinements described above, the users have access to four drop down menus: the Topic Options menu lets users merge topics, split a topic (Fig 3(a)), or rename a topic (Fig 2(b)). The Word Options menu allows users to add words to stopwords, remove words from a topic (Fig 3(d)), or add words to a topic . The Document Actions menu allows users to remove a document from the corpus, add documents to a topic, or remove documents from a topic (Fig 3(c)). The Model Options menu allows users to evolve the model (iterate the Gibbs sampler), save the model, load a model, or reinitialize the model, which loads the corpus anew and begins the process from scratch with a user defined number of topics. The changes made to the model take place immediately after the user specifies the action.

4.3 Implementation

This project was implemented in Python. However, since there is need for an end user to use this without significant delay, the Gibbs sampler was implemented in Cython, and compiled into C code, which is then called from Python. The functionalities of the refinements are implemented either in Python using NumPy, or in Cython if deemed appropriate. The front End itself was implemented using Plotly's Dash tool [14]. The code for this project can be found at https://github.com/AmFamMLTeam/hltm_welda.

Figure 1: Work flow of human-in-the-loop topic modeling with WELDA.

5 EMPIRICAL EVALUATION

While extensive research has been done on the variants of LDA topic discovery models, how to evaluate topic discovery remains an open research area for decades [7, 17, 31]. Most proposed metrics fall into one of the following categories: bench marking against document classification tasks with known document topic assignments, or some measure of human interpretability of the topics. The latter is typically measured using pointwise mutual information (PMI) [23] or more sophisticated topic coherence [27]. Therefore in the following context we evaluate proposed HL-TM WELDA in some established metrics. We also demonstrate an example of how insurance domain experts can use HL-TM WELDA to perform previously tedious and manual business topic discoveries from a vast amount of unstructured text.

5.1 Topic Discovery on 20 Newsgroups

We first benchmark our proposed HL-TM WELDA initialization results against standard LDA and WELDA using the 20 Newsgroups [16] data set, a collection of 18828 newsgroup documents partitioned into 20 different categories. Each of the categories has an approximately even number of document. When applying WELDA, we calculated word embeddings using a Word2Vec [22] model, pre-trained on Wikipedia, for each word in each document. Table 2 shows the result of comparing HL-TM WELDA, WELDA, LDA 's initialization results in a number of standard clustering evaluation

metrics: purity [21], rand index [1], normalized mutual information (NMI) [25], homogeneity, completeness, V-measure [28] and Fowlkes-Mallows (F-M) score [3]. Among those metrics, the purity score can be thought of as follows: For each cluster, count the number of data points from the most common class in said cluster. Now take the sum over all clusters and divide by the total number of data points. It is equivalent to an aggregated classi cation accuracy of the 20 news classes when the class of clusters was assigned by majority voting. Normalized mutual information is the expectation of PMI [23], a popular measure for topic modeling.

	purity	Rand idx.	NMI	homog.	completeness	V-measure	F-M
LDA	0.387	0.255	0.342	0.339	0.351	0.345	0.298
WELDA	0.386	0.242	0.352	0.349	0.360	0.354	0.285
HL-TM WELDA	0.454	0.266	0.385	0.383	0.392	0.387	0.306

Table 2: 20 Newsgroups topic evaluation metrics for LDA, WELDA initialization and WELDA after human in the loop iterations

	WELDA		HL-TM WELDA	
bsbl act.	725	14	481	193
hcky act.	685	132	8	813
	bsbl pred.	hcky pred.	bsbl pred.	hcky pred.

Table 3: Confusion matrices for classes rec.sport.baseball and rec.sport.hockey for both WELDA and HL-TM WELDA. Topic assignment from a cluster is given by majority voting.

The rst two rows in Table 2 show WELDA's initial results are comparable to those obtained by LDA. The third row, however, shows that our proposed HL-TM WELDA is consistently better than the other two methods across all metrics. This seems to empirically indicate that the domain knowledge provided by the human during the topic discovery process plays a signi cant role into the formation of the topics.

One of the improvements from LDA and WELDA to HL-TM WELDA comes from being able to interactively separate words and documents from one initially formed cluster into two separate clusters. Table 3 shows confusion matrices for both WELDA and HL-TM WELDA across the topics rec.sport.baseball and rec.sport.hockey . This shows that HL-TM WELDA can help to disambiguate the two topics. The accuracy of LDA and HL-TM WELDA for these two topics is 0.55 and 0.87 respectively.

5.2 Topic discovery on insurance claims rst notice of loss (FNOL)

5.2.1 Loss cause discovery from FNOL.

Insurance business processes usually generate a large amount of information-rich unstructured text. Being able to understand and distill knowledge and insights from those texts can provide valuable business opportunities. One such example is to identify loss causes from the insurance claims' rst notice of loss (FNOL). FNOL is a piece of unstructured text brie y describing the situation of the accidents when the claim is led. The average length of FNOL is 52 words. Here is an example of FNOL description of a recreational vehicle (RV) accident:

> The front right tire on the trailer blew out and broke apart causing damage to the wheel well and possibly to the wiring on the electronic brakes for that wheel.

Nowadays insurance product analysts go through a manual loss cause discovery process using keywords search and regex pattern match. When the set of FNOL text becomes large, this process easily become tedious and can miss topics in the corpus.

5.2.2 HL-TM WELDA loss cause discovery from FNOL.

In the following context we show how an insurance product analyst used HL-TM WELDA to interactively discover and re ne loss causes from RV claim FNOL descriptions. The analyst initialized the HL-TM WELDA tool with 7287 RV FNOL descriptions and set seven topics i.e. loss causes (Figure 2(a)). She determined the number of topics and would change number of topics later using split topic or merge topic operations. After reading the top 10 ranked FNOL documents in a topic, the analyst did a majority voting to nd the appropriate loss cause name of this topic. In this example, after the model initialization, the analyst could easily identify and rename ve relatively coherent topics into loss causes (Figure 2 (b)). The following list shows the initial loss cause identi ed and their FNOLs breakdowns:

- Collision with stationary objects. Seven out of ten WELDA top ranked documents belongs to this topic.
- Tire blew out. Five out of ten WELDA top ranked documents belongs to this topic.
- Hail Damage. Six of ten WELDA top ranked documents belongs to this topic.
- Awning damage from the wind. Nine of ten WELDA top ranked documents belongs to this topic.
- Roof leak or water damage. Eight out of ten WELDA top ranked documents belongs to this topic.
- TOPIC 1 is a mix of documents belonging to collision with stationary objects, awning damage, theft/vandalism, collision with vehicles etc.
- TOPIC 4 is a mix of collision belonging to collision with stationary objects, collision with other vehicles, collision with animal, water damage etc.

To re ne topics with mixed themes, e.g. TOPIC 4, the analysts usually would perform three actions: re assign documents to other already de ned topics (Figure 3 (a)); remove documents ; or split a new topic by choosing representative seeding keywords (Figure 3(b)). In this example, a new loss cause COLLISION WITH VEHICLE was split from TOPIC 4, by seeding keywords dent, turn, left, right, lane To further tweaking the quality of those topics, the analyst can remove word from topic (Figure 3 (c)) and remove documents from topics (Figure 3 (d)). The HL-TM WELDA UI also provides batch removal options forcing the underlying model to forget the words and documents assignment faster.

After removing documents and splitting topics from TOPIC 4, the analyst discovered that the model surfaced a few documents about animals causing damages. Considering animal damage topic is pretty novel, she kept removing documents to form a topic about loss cause Animal. Similarly the analyst is able to re ne TOPIC 1 into loss cause THEFT/VANDALISM. Using the human-in-the-loop process, the analyst is able to prune away the majority documents

but to keep the novel observations for the model to update the distributions. Figure 4 shows that the analyst is able to interactively form the final eight loss causes.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have developed a state of the art framework for topic discovery that incorporates both human-in-the-loop feedback functionalities as well as utilizing word vector models trained on a domain-specific corpus. The resulting topic model can thus be guided by a domain expert as well as be informed by the knowledge encoded in a word vector model specific to the insurance domain. Our quantitative evaluations on 20 newsgroup data sets has shown our proposed WELDA is comparable to the traditional LDA in a number of clustering metrics. The evaluation also showed HL-TM WELDA outperforms both WELDA and LDA, and the improvements primarily stem from the human in the loop interactions with the underlying statistical topic model. Our qualitative evaluation walked through an example of how an insurance used this tool to discover topics from the insurance domain unstructured text.

In addition to continue developing and adding features to the deployed version of this tool, future work includes investigating how to leverage a knowledge base in the form of a knowledge graph to help inform the human-guided topic discovery process, investigating the possibility of discovering topic hierarchies as investigated in [5] in a similar but different context, and exploring the use of contextualized text embedding representations [10].

REFERENCES

[1] L. H. . P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[2] K. Batmanghelich, A. Saeedi, K. Narasimhan, and S. Gershman. Nonparametric spherical topic modeling with word embeddings. In *ACL*, pages 537–542, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[3] E. B.Fowlkes and C. L.Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[5] J. Bockhorst, D. Conathan, and G. Fung. Knowledge graph-driven conversational agents. 2019.

[6] S. Bunk and R. Krestel. Welda: Enhancing topic models by incorporating local word context. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, JCDL '18, page 293–302, New York, NY, USA, 2018. Association for Computing Machinery.

[7] J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, pages 288–296. Curran Associates, Inc., 2009.

[8] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19:1992–2001, 2013.

[9] R. Das, M. Zaheer, and C. Dyer. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, Beijing, China, July 2015. Association for Computational Linguistics.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[11] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[12] E. Hoque and G. Carenini. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 169–180, New York, NY, USA, 2015. ACM.

[13] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive topic modeling. *Machine Learning*, 95(3):423–469, Jun 2014.

[14] P. T. Inc. *Collaborative data science*. Plotly Technologies Inc., Montreal, QC, 2015.

[15] D. Jin, J. Huang, P. Jiao, L. Yang, D. He, F. Soulie-Fogelman, and Y. Huang. A novel generative topic embedding model by introducing network communities. In *The World Wide Web Conference*, WWW '19, pages 2886–2892, New York, NY, USA, 2019. ACM.

[16] T. Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In D. H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, pages 143–151. Morgan Kaufmann, 1997.

[17] J. H. Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In G. Bouma and Y. Parmentier, editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 530–539. The Association for Computer Linguistics, 2014.

[18] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.

[19] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012.

[20] T. Y. Lee, S. Alison, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105, 03 2017.

[21] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[23] D. Newman, S. Karimi, and L. Cavedon. External evaluation of topic models. In *in Australasian Doc. Comp. Symp., 2009*, pages 11–18, 2009.

[24] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson. Improving topic models with latent feature word representations. *ACL*, 3:299–313, 2015.

[25] X. V. Nguyen, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In A. P. Danyluk, L. Bottou, and M. L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 1073–1080. ACM, 2009.

[26] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[27] M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM.

[28] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In J. Eisner, editor, *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 410–420. ACL, 2007.

[29] A. Smith, V. Kumar, J. L. Boyd-Graber, K. D. Seppi, and L. Findlater. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In S. Berkovsky, Y. Hijikata, J. Rekimoto, M. M. Burnett, M. Billinghurst, and A. Quigley, editors, *Proceedings of the 23rd International Conference on Intelligent User Interfaces, IUI 2018, Tokyo, Japan, March 07-11, 2018*, pages 293–304. ACM, 2018.

[30] A. Smith, V. K. Vijay, J. L. Boyd-Graber, K. D. Seppi, and L. Findlater. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *IUI*, 2018.

[31] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. M. Mimno. Evaluation methods for topic models. In A. P. Danyluk, L. Bottou, and M. L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 1105–1112. ACM, 2009.