Word Embedding Based Extension of Text Categorization Topic Taxonomies

Tobias Eljasik-Swoboda[1], Felix Engel[2], Michael Kaufmann[3], and Matthias Hemmje[2]

[1] University of Hagen, Universitätsstraße 47, 58084 Hagen, Germany
[2]FTK e.V. Forschungsinstitut für Telekommunikation und Kooperation, Martin-Schmeißer-Weg 4, 44227 Dortmund, Germany
[3] Lucerne University of Applied Sciences and Arts, Technikumstrasse 21, 6048 Horw, Switzerland
Tobias.Swoboda@fernuni-hagen.de
{Felix.Engel, Matthias.Hemmje}@ftk.de
m.kaufmann@hslu.ch

Abstract. Collaborative interdisciplinary research has the added difficulty, that researchers from different fields have different backgrounds and employ heterogeneous technical vocabularies. Certain problems could have already been solved in one field, but the solution is described in such a fashion, that it is difficult for researchers from another field to understand, yet alone to know the correct terms to search for. Text categorization (TC) is the act of automatically placing text into content-based categories. These categories can be interrelated forming hierarchical taxonomies of knowledge. Different from classic querying-based information retrieval (IR), TC-based IR allows for an exploration of topics without prior knowledge about them, by inspecting the individual topics and related documents within the taxonomies. TC also plays a major role in argumentation mining (AM), the automated extraction of arguments from large quantities of text. In AM, TC is used to identify argument structures within analyzed texts. Another potential use for TC in AM is the restriction of data sources to relevant topics because AM in too-large text corpora can be prohibitively time consuming. As mankind's knowledge constantly expands it is logical to conclude, that the taxonomies organizing this knowledge must expand as well. We propose a method to aid in extending existing topic taxonomies by using word embeddings. These extended topic taxonomies can then be used in the categorization of texts, and to filter argument-extraction sources. We additionally outline an alternative usage of these techniques in argumentation mining.

Keywords: Taxonomies, word embedding, text categorization

1 Introduction and Motivation

The main goal of argumentation mining (AM) is to automatically extract arguments from generic texts to provide structured data for computational models of argument and reasoning engines. To accomplish this goal, argumentation models are used. These models form parts of individual arguments (Lippi and Torroni, 2016). According to

Habernal and Gurevych (2015), the prevailing model of arguments in AM is that of a discourse structure consisting of several argument components, such as premises and claims. Text categorization (TC) is the act of automatically assigning texts of arbitrary length to a predefined set of categories (Sebastiani, 2002). When modeling sentences within the mined text corpora as texts and argument components—using, for example, premises and claims as categories—TC is the foundation for a plethora of AM systems (Mochales and Moens, 2011; Feng and Hirst, 2011; Rooney et al., 2012; Stab and Gurevych, 2014).

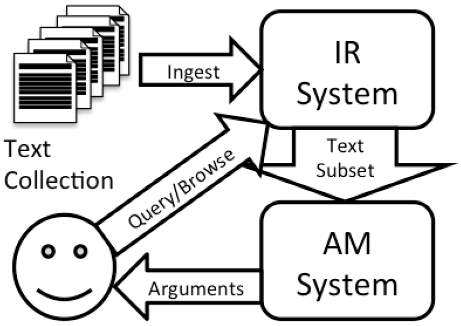

Fig. 1: Information retrieval (IR) before argumentation mining (AM).

ArgumenText is a practical implementation of an AM engine (Stab et al., 2018). It employs a two-step mechanism in which a large collection of documents (http://commoncrawl.org/, in Stab et al.'s experiment with 683 GiB) is first indexed into an information retrieval (IR) engine. The user can then query the engine using search terms. The resulting subset of documents is subsequently mined for arguments (see Figure 1). This is done in order to reduce computation time because AM on this scale takes too much time with access only to ordinary hardware. In order to query the engine, the user must know the exact search terms to be used.

Having a taxonomy of topics could allow the browsing of different facets of topics without prior knowledge about their exact structure and common sub-topics. This way, TC could be used as an alternative to the regular querying-based IR engine, allowing browsing-based topic exploration. Such taxonomies also directly benefit collaborative interdisciplinary research. Our research originates from the RecomRatio project. The goal of RecomRatio is to provide medical professionals with treatment recommendations that were extracted from current medical literature, arguments for or against these treatments, and the analyzed medical literature itself. Therefore our experiments have a strong medical focus. Before TC can be performed, one needs a set of categories, *C*. This is obviously given for argument structures but could be lacking when one models a topic taxonomy for exploration. The aim of our work is to help in the creation of such a topic taxonomy by suggesting extensions to an existing proto-taxonomy (see Figure 2). Because TC usually works in a supervised-learning fashion, one also requires example text-to-category assignments. This need has been remediated in newer unsupervised TC techniques (Dai et al., 2017; Eljasik-Swoboda et al., 2018). These techniques are based on word embeddings (see section 2). Following Dai et al.'s and Eljasik-Swoboda et al.'s examples, we propose a method to suggest taxonomy extensions for existing topic taxonomies using unsupervised machine learning while processing no data other than a large collection of example texts and an existing initial topic taxonomy. Given natural language texts about the topic and an initial proto-taxonomy as input, our system will then suggest sub-topics for a given topic in this taxonomy. For example, when analyzing texts about *melanoma*, the system will suggest sub-topics for *melanoma* in a taxonomy tree that models different diseases. This example taxonomy could model *cancer* as a family of diseases and have *melanoma* as a sub-category of *cancer*.

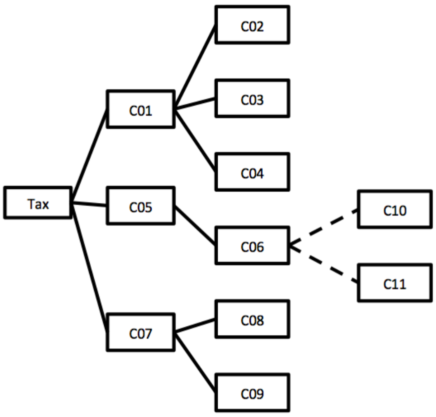

Fig. 2: Topic taxonomy extension.

Even though plenty of resources are available for the medical domain, we limit ourselves to this because these resources might not be available for emerging cutting-edge topics, as medical researchers are likely to first describe them using natural language before making them machine readable in any fashion or agreeing upon their technical vocabulary definitions (Nawroth et al., 2018). This makes our approach transferable to any natural language and uniquely suited for emerging knowledge domains because, during the adoption phase of TC for any application, additional—especially manually compiled—information resources are difficult to obtain. Our contribution is two-fold. First, we propose a novel unsupervised method to help in the introduction of TC as an IR method for AM or any other application by proposing new categories. Second, we analyze and discuss what influences the effectiveness of our system—such as, for example, the utilized word embedding algorithm.

2 State of the Art and Related Work

In order to model any topic relationships, one needs a way to model the semantics of individual terms. Ontologies are manually created encodings of semantics. They are commonly used in all types of natural language processing applications. Given the high amount of work put into developing ontologies, the ontologies are very precise in capturing the semantic understanding of their creators (Busse et al., 2015). Their drawback is that they need to be manually created. Blei et.al (2003) proposed another fundamental approach to capturing semantics for words. Their latent dirichlet allocation (LDA) statistically splits documents into topic distributions and divides topics into term distributions. Each term is assigned a topic vector that comprises its probability of being part of each topic. LDA is also referred to as topic modeling. Here, terms are regarded as similar if they occur in the same document. Camiña (2010) described multiple methods to generate taxonomies based on term similarities in LDA topic distributions. The same is true for Kashyap et al.'s TaxaMiner experimentation framework for automated taxonomy bootstrapping. Even though these methods are appealing, we pursuit a different goal in trying to extend an existing proto-taxonomy instead of starting from scratch.

Another intriguing approach lies in word embeddings. These are unsupervised learning methods that can capture semantic relatedness by analyzing large texts or concatenations of multiple smaller texts. Word2Vec is a prominent implementation of word embeddings that Mikolov et al. (2013) developed. Word2Vec consists of two algorithms, continuous bag of words (CBOW) and skip-gram. Both produce high-dimensional coordinates for every word and operate by optimizing the cosine similarity between each word. In CBOW, the similarity of terms that are surrounded by the same context terms is maximized. In skip-gram, the similarity of the context terms surrounding the same central terms is optimized. Words are considered to surround a term if they are in a context window of n words before or after the term. Using this pattern, semantic relatedness becomes encoded by similar offsets that capture multiple dimensions of meaning. To the best of our knowledge, this has not been observed in LDA-based term vectors. A reason for that can be the higher granularity of word embeddings regarding what terms are in the other words' contexts.

Habernal and Gurevych (2015) utilized this in the context of AM by creating clusters of terms commonly used in arguments in order to support the annotation of arguments within text. Fu et al. (2014) also used word embeddings to extract hypernym/hyponym relationships between terms in order to create an ontology. Their experiments suggest that a simple hypernym/hyponym vector offset does not exist; rather, one offset exists per class of terms. For example:

v(shrimp) - v(prawn) ≈ v(fish) - v(goldfish) and v(laborer) - v(carpenter) ≈ v(actor) - v(clown) but v(laborer) - v(carpenter) ≉ v(fish) - v(goldfish).

Our objective is similar to that of Fu et al. (2014). Instead of extracting hypernym/hyponym relationships between terms, we attempt to extend topic taxonomies with sub-categories. These sub-categories are not necessarily hyponyms, as they could also cover certain aspects of their parent categories. As we limit ourselves to only the existing text and initial taxonomies, word embeddings are an optimal foundation for our method. As previously mentioned, our topical focus is in the medical domain. A cornerstone of medical literature is PubMed, the National Institutes of Health's U.S. National Library of Medicine database (U.S. National Library of Medicine, 1996). PubMed includes a querying-based search engine and abstracts for most indexed articles. The articles themselves are stored elsewhere, with their references and DOIs available in PubMed. Additionally, articles are annotated with Medical Subject Headings (MeSH) (U.S. National Library of Medicine, 1999). MeSH is updated annually, currently defines 28,378 medical topics, and organizes these topics 58,025 times in 16 topical taxonomies such as *anatomy* and *diseases*. In these taxonomies, every topic has one or multiple paths from the taxonomy root to its entry. Even though these taxonomies form directed acyclic graphs (DAGs), some topics are listed multiple times in the same taxonomy.

Kaufmann et al. (2017) created the big-data management canvas (BMDC). The fundamental insight is that the aim of any big-data project is the effectuation: the creation of a benefit through the analysis of big amounts of data. The same is true for any data- or text-mining endeavor. In order to not lose sight of this, the BDMC planning method splits endeavors into five main fields of activities. These fields form a loop of activities going from the datafication (which is the capturing of data for later analysis) to the said effectuation. These are further split into a business aspect and a technology aspect. The business aspect describes and plans what should be done whereas the technology aspect describes and plans how it should be implemented. We used this method during the planning and modeling of our system.

3 Model and Implementation

As mentioned, we organized this research using the BDMC that Kaufmann et al. (2017) proposed. The following sub-sections reflect the BDMC's fields of activity. This illustrates the workflow we propose for the extension of topic taxonomies. We named our system Taxonomy Extension system for Emerging Knowledge (TEEK), as its primary task is to capture emerging topics for usage in TC. We used the BDMC to structure the creation of our prototype as well as the performed evaluation experiments.

3.1 Datafication

The BDMC defines datafication as the act of transforming real-world events and properties into usable data. It also closes the loop to the effectuation, as every effectuation influences the world we live in and hence creates new data to capture. In our envisioned application, the relevance feedback provided by the domain experts curating the taxonomy is the datafication of this endeavor. If a domain expert agrees with the system and adds a category to the system, the available taxonomy changes. The datafication of our experiments is performed with the evaluation of the proposed categories as described in section 3.4.

3.2 Data Integration

The BDMC field of data integration describes which data is used, how it is obtained, and how it is centrally managed and stored. As mentioned before, our system works on taxonomies and text files about a given knowledge domain. We performed experiments for the medical terms *neoplasms* (cancers/tumors), *melanoma*, *leukemia*, *Herpesviridae*, and *Simplexvirus*. Each of these terms has one or multiple entries in a MeSH taxonomy. *Melanoma*, *leukemia*, and *neoplasms* are part of the *diseases* taxonomy whereas *Simplexvirus* and *Herpesviridae* are part of the *anatomy* taxonomy. *Simplexvirus* is a descendant of *Herpesviridae* whereas *melanoma* and *leukemia* are descendants of *neoplasms* within MeSH.

The finished system will have access to a multitude of documents from which it can learn the relationships between terms in order to propose new topics. For our prototype, we simulate this using PubMed. We queried PubMed for each of the above-mentioned terms and used the export-to-XML function in order to download all metadata and abstracts for a given topic. Word2Vec requires lowercase texts without special characters. Because the resulting XML files were up to 31.01 GiB in size (neoplasms), we implemented a buffered XML parser in Java that extracts the abstracts from all articles in the individual result sets and stores them into simple text files, removing all special characters. For easy integration in multiple applications, we packaged the original C implementation of Word2Vec (https://github.com/tmikolov/word2vec), into a Docker container, which we used to run CBOW and skip-gram on these extracted text files. This means that we have two word embedding files for each PubMed search term to experiment on. We parameterized them to have 200 dimensions and use a five-word (before and after) context window.

3.3 Data Analytics

The data analytics field describes how the available data is analyzed. As previously stated, the available data is a set of word embeddings and an initial taxonomy. Every taxonomy—$T = \{C, E, L\}$—has a set of categories—C—and a set of labels for each category, L. Additionally, the set of edges, E, between the categories form a DAG with $r \in C$ at its root. The labels consist of one or multiple words.

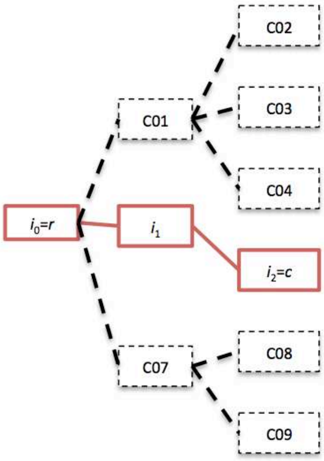

Fig. 3: Path from root to category c.

Word embeddings are high-dimensional vectors for all terms that the algorithm encounters during training. We denote them as *v*(word). This way, a word embedding vector can represent every category with single-word labels. If the label of one category consists of multiple words, we compute its vector representation by calculating the arithmetic mean of all the individual word vectors. Because T forms a DAG, every $c \in C$ has a path $i_0, \dots, i_n$, where $i_0 = r$ and $i_n = c$. With the word embeddings, every node has a representation in vector space. The rest of the taxonomy is ignored (see Figure 3).

The task of suggesting new sub-categories for c is essentially that of extending the path to $i_0, \ldots, i_n, i_{n+1}$. Our approach for TEEK is to compute the most-likely next category label vector by using the information provided by the existing path $i_0, \ldots, i_n$. Once this vector is computed, the 10 closest terms to this next vector in word embedding space are calculated using cosine similarity. It is noteworthy that the closest term in all our experiments is the label of c. The system therefore creates nine suggestions per category.

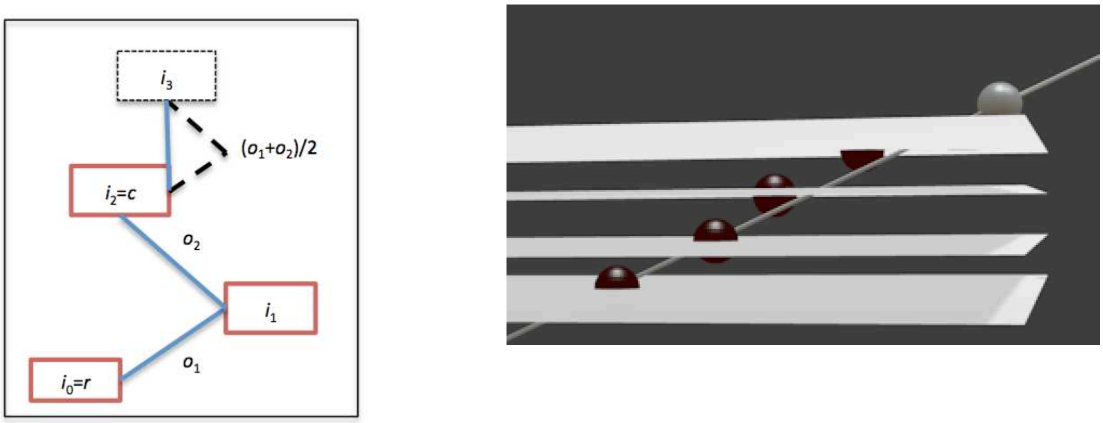

Fig. 4: AVSP and RVSP illustration and comparison.

We experimented with two possible variations for this task. The first variation begins with the computation of the offsets between individual categories on the path: $o_j = v(i_j) - v(i_{j-1})$. In the next step, it adds the average offset between all categories on the path to the vector of i_n:

$$(i_{n+1}) = v(i_n) + \frac{1}{n}\sum_{j=1}^{n} o_j \tag{1}$$

This equation essentially adds the arithmetic mean of the individual offsets to the last vector. We therefore refer to it as arithmetic vector-stream predictor (AVSP). Figure 4 portrays this approach in an example two-dimensional word embedding space.

The second variation applies linear regression to the problem of finding the hyperplane closest to all $v(i)$. Using the following equations, this hyperplane is expressed as function of path index j:

$$\bar{j} = \frac{n^2+n}{2n+2} \tag{2}$$

$$\bar{v} = \frac{1}{n}\sum_{j=0}^{n} v(i_j) \tag{3}$$

$$b = \frac{\sum_{j=0}^{n}(j-\bar{j})*(v(i_j)-\bar{v})}{\sum_{j=0}^{n}(j-\bar{j})^2} \tag{4}$$

$$a = \bar{v} - b * \bar{j} \tag{5}$$

This way, the next word embedding vector is found through the following equation:

$$v(i_{n+1}) = a + b * (n+1) \tag{6}$$

Because this is standard linear regression applied to vectors instead of scalars, we named our second approach regression vector-stream predictor (RVSP). Figure 4 portrays this using two-dimensional word embedding spaces. The word embedding space is shown for each term with the distance from the root of the taxonomy forming an additional dimension. The dark-red spheres represent the individual categories, the planes symbolize the word embeddings that the categories are in, and the line represents the hyperplane closest to all these points. The white sphere represents the suggested categories, as it extends the taxonomy-depth dimension by one.

Our prototype implements these two methods in Java after reading word embeddings as text files and the taxonomy as an XML file from the file system. Both approaches essentially create a direction in the word embedding space that reflects the direction of the path from the taxonomy root to the individual topic category. This captures Fu et al.'s (2014) finding regarding the lack of a common hyponymy direction between different classes of terms. Here we use the available information provided by the taxonomy structure to discern different dimensions appropriate for each topic category.

3.4 Data Interaction and Data Effectuation

The data interaction field describes how users interact with the data in order to benefit from the data effectuation. In our case, a domain expert can review the category suggestions before accepting them for usage in the topic taxonomy. After a suggested topic is accepted, a TC algorithm can assign content to this category. This review and acceptance component will be implemented through a Web interface that will show the user the top nine suggestions to extend for a given taxonomy. This forms a type of relevance feedback for the suggested categories. For our prototype, the results are stored in a Microsoft Excel file to ease their review and validation by the medical professionals that support us in this project (see section 3.2). The goal of our proposed system is the automatic suggestion of additional topic categories in order to extend an existing taxonomy. A TC-oriented IR system subsequently uses these categories to allow uninformed IR. This uninformed IR can then used to narrow down the source material used for AM. This narrowing down is crucial for performing AM in a timely fashion.

4 Evaluation and Result Interpretation

The purpose of our evaluation is to discuss the usefulness of suggested sub-categories for the given topic. Because our method can be parameterized differently, we can investigate the effect of the selected parameters on the results. As mentioned before, we use MeSH and PubMed as data sources for our experiments. The assessed topics are not leafs in MeSH but rather are inner nodes that already have a set of sub-topics in MeSH. The already existing sub-topics are hidden from our algorithm. This allows for four types of true positive results for our system: First, suggested sub-categories that are actual existing sub-categories of the investigated topic and, second, suggested categories that are not already sub-categories in MeSH but would make sense as subcategories according to publically available medical sources. Examples for these findings are *myeloblastic* as a sub-class of *leukemia* and *lentiginous melanoma*. *Myeloblastic leukemia* and *lentiginous melanoma* are types of their diseases that have been described in literature but are not modeled as sub-categories in MeSH. We published all detailed results including references for potentially meaningful sub-categories at: https://github.com/SirTobiSwobi/TEEKeval . We regard different spellings of the category name as correct sub-category suggestions and therefore TP because our system correctly interpreted them as types of the category. Our system correctly captured, that a misspelling of *leukemia* (like *leukeamia*) must be some kind of *leukemia* because experts wrote about in the same way. We interpret the plural of a category as fourth type of TP result, because the system correctly captured that sub-categories are different types of the original category. For example, it recognized, that the sub-categories of *melanoma* are (different types of) *melanomas*. Albeit plural- and different spelling results are not directly helpful in extending an existing taxonomy, they aid in comparing the effectiveness of different parameters and approaches. This allows us to compare which word embedding algorithm and extrapolation method produce better results. Additional insights can be gained by using different source material for the word embeddings. *Leukemia* and *melanoma* are descendants of *neoplasms*. Therefore, we can compare the performance of word embeddings generated through the PubMed abstracts to the more specific search term or through the larger amount of abstracts using the more general search term. The same is true for *Herpesviridae* and *Simplexvirus*.

Although *melanoma* occurs three times in 2018's MeSH, all other examined medical terms have only one entry. This allows another investigation about how the path length influences the performance of the system. Of these three entries, two entries are six steps removed from the taxonomy root whereas one entry is only four steps removed. The effectiveness of IR systems is usually measured in *precision* and *recall* (Sebastiani, 2002). They are not directly applicable because we do not perform information retrieval but attempt to extend topic taxonomies. To compare different word embedding algorithms, source material and path lengths for individual terms, we use a modified version of *precision*. Results that are on the path between the root and the term as well as other relative terms and completely unrelated terms are treated as False Positive (FP). Because we know the existing proto-taxonomy, results on the path or other relative terms could be filtered from the result set, so that the system instead outputs the next closest term in word embedding space. We decide not to do this for the sake of comparing different parameters. Precision is the ratio between TP and FP with 1.0 meaning only TP and 0.0 meaning only FP. For the recall measure, one needs to know all possible correct relevant sub terms, which nobody in our team did. Therefore, we only measure precision for our system. Table 1 contains the precision values for all our performed experiments.

Table 1. Experimental results.

			AVSP		RVSP	
Term	Depth	Abstracts	CBOW	skip-gram	CBOW	skip-gram
Neoplasms	1		44%	44%	56%	33%
Melanoma 1	6	Low level	11%	11%	0%	0%

Melanoma 2	6	Low level	11%	11%	0%	0%
Melanoma 3	4	Low level	33%	22%	67%	67%
Melanoma 1	6	High level	44%	56%	33%	33%
Melanoma 2	6	High level	44%	56%	33%	33%
Melanoma 3	4	High level	67%	56%	67%	56%
Leukemia	3	Low level	67%	33%	56%	11%
Leukemia	3	High level	56%	89%	78%	89%
Herpesviridae	3		22%	56%	44%	56%
Simplexvirus	5	Low level	22%	11%	0%	0%
Simplexvirus	5	High level	22%	0%	22%	11%

On average, the AVSP (37%) and the RVSP (35%) performed almost equally as well. The RVSP found no TP in 6 out of 24 experiments. This only happened to the AVSP in one experiment. The RVSP is almost on par, because it delivered better results in other experiments. When comparing the average precision of the CBOW (38%) and skip-gram (35%) word embedding algorithms, both delivered comparable results no matter the extrapolation method. Albeit CBOW delivered on average slightly better results, skip-gram produced the best single result (89%).

Training the word embeddings on larger (high-level) text collections (E.g. Neoplasms instead of melanoma) had the biggest impact on performance. Low-level representations only yielded an average effectiveness of 22%, providing almost all cases in which no TP were found, while high-level representations had an average effectiveness of 47%. Metaphorically speaking this means that the more the system "knows", the better it is at generating sub-category suggestions. Another influencing factor is the depth of the investigated term within the taxonomy. With depth 1, the average precision was 44%, with depth 3 60%, and depth 4 54%. With depth 5, only 11% precision was achieved while depth 6 on average generated 24%. This means that with a depth>4, less than half the precision that with depth < 4 was achievable. We see two reasons for this behavior: The deeper a category is in a taxonomy, the more intermediate steps the system can use for extrapolation of sub-categories. On the other hand, the more specialized a topic is, the less likely there are many sub-topics.

5 Conclusions and potential AM usage

Our work shows a new way to extend existing topic taxonomies by using no other information than texts about the knowledge domain and an initial topic taxonomy. Therefore, domain experts do not need to perform any manual effort besides accepting suggested categories. As initially explained, text categorization using topic taxonomies supports uninformed information retrieval in any application and can specifically be applied for narrowing down documents before AM. To do so, TC systems require appropriate topic taxonomies. Our prototype aids in creating these taxonomies with only little labeled data. This directly benefits ArgumenText, RecomRatio, and other AM systems. Besides this filtering application, another potential utilization for this approach in AR is the detection of pros and cons for individual terms. To utilize this system as such, one could construct a taxonomy of topics or simply use an existing one such as MeSH. Afterward, known pros are manually modeled as sub-categories for the topic categories. For each new topic-pro-leaf, the average offset or regression-based extrapolation vector can be computed. Adding this vector to $v(i_{n-1})$ instead of $v(i_n)$ would allow a user to find other pros for the leaf topic. The same can be done for cons. Using this technique, potential features for spotting pros and cons for topics can be extracted. These can then be used in AM TC or by a user to manually come up with arguments for or against something that are not already written down in the texts mined for arguments. In addition to describing a new way to extend taxonomies, we investigated how different parameters influence the approach. Albeit CBOW slightly outperforms skip-gram, the latter achieved the best individual results. Similarly, the AVSP slightly outperformed the RVSP on average. The RVSP however had many more results without TPs. Upon investigation, we found that it delivered many terms describing topics on the path from the root to the term in question. As mentioned, these can easily be filtered in future works. The strongest influence on performance comes from the texts that the word embeddings are based on. The more text about more general topics is analyzed, the better the system performs. This means that the hypothetical best results would come from word embeddings that are generated through the use of all abstracts on PubMed. Due to resource constraints, we were not able to practically test this. The taxonomy depth of the extended category also plays an ambiguous role: The further the category is removed from the root, the more intermediate steps can be used in the analysis. However, the more specific a topic is, the less likely it is to have more sub-topics.

Acknowledgements

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project "Rationalizing Recommendations", Grant Number 376059226, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

1. David M. Blei, Andrew Y. Ng, and Michael I. Jordan, 2003. Latent dirichlet allocation. In *Journal of machine Learning research*, pages 993-1022.
2. Johannes Busse, Bernhard Humm, Christoph Lübbert, Frank Moelter, Anatol Reibold, Matthias Rewald, Veronika Schlüter, Bernhard Seiler, Erwin Tegtmeier, and Thomas Zeh, 2015. Actually, What Does “Ontology” Mean? A Term Coined by Philosophy in the Light of Different Scientific Disciplines. In *Journal of Computing and Information Technology (CIT),* 23, 1, pages 29-41. https://doi.org/10.2498/cit.1002508
3. Steven L. Camiña. A comparison of Taxonomy Generation Techniques Using Bibliometric methods: Applied to Research Strategy Formulation, Composite Information Systems Laboratory (CISL) Working Paper 2010-01, Massachusetts Institute of Technology, 2010
4. Xiangfeng Dai, Marwan Bikdash, and Bradley Meyer, 2017. From social media to public health surveillance: Word embedding based clustering method for twitter classification. In *Proceedings of SoutheastCon,* pages -7, https://doi.org/10.1109/SECON.2017.7925400
5. Tobias Eljasik-Swoboda, Michael Kaufmann, and Matthias Hemmje, 2018. No Target Function Classifier Fast Unsupervised Text Categorization Using Semantic Spaces. In *Proceedings of DATA 2018*, https://doi.org/10.5220/0006847000350046

6. Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 987–996, Portland, Oregon. Association for Computational Linguistics.
7. Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu, 2014. Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 119-1209.
8. Ivan Habernal, and Iryna Gurevych. 2015. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2127-2137.
9. Vipul Kashyap, Cartic Ramakrishnan, Christopher Thomas, Amit P. Sheth, 2005, TaxaMiner: An Experimentation Framework for Automated Taxonomy Bootstrapping. International Journal of Web and Grid Services, 1 (2), pp. 240-266, available online: http://corescholar.libraries.wright.edu/knoesis/744
10. Michael Kaufmann, Tobias Eljasik-Swoboda, Christian Nawroth, Kevin Berwind, Marco Bornschlegl, and Matthias Hemmje, 2017. Modeling and Qualitative Evaluation of a Management Canvas for Big Data Applications. In *Proceedings of DATA 2017*, https://doi.org/10.5220/0006397101490156
11. Marco Lippi, and Paolo Torroni. 2016. Argumentation Mining: State of the Art and Emerging Trends. In *ACM Transactions on Internet Technology (TOIT)*, 16.2, pages 10-35.
12. Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, 2013. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746-751.
13. Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. In *Artificial Intelligence and Law*, 19(1), pages 1-22.
14. Christian Nawroth, Felix Engel, Tobias Eljasik-Swoboda and Matthias Hemmje, 2018. Emerging Named Entity Recognition for Clinical Argumentation Support. In *Submitted to Proceedings of DATA 2018*, https://doi.org/10.5220/0006853200470055
15. Niall Rooney, Hui Wang, and Fiona Browne. 2012. Applying kernel methods to argumentation mining. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pages 272–275. Association for the Advancement of Artificial Intelligence.
16. Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. In *ACM computing surveys (CSUR)*, 34.1, pages 1-47.
17. Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar, October. Association for Computational Linguistics.
18. Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArgumenText: Searching Arguments in Heterogeneous Sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21-25.
19. U.S. National Library of Medicine. 1996. PubMed. Available at: https://www.ncbi.nlm.nih.gov/pubmed/ (accessed: May 28, 2018)
20. U.S. National Library of Medicine. 1999. Medical Subject Headings. Public Domain Information. Available at: https://www.nlm.nih.gov/mesh/ (accessed: May 29, 2018)