Offering Answers for Claim-Based Queries: A New Challenge for Digital Libraries

José María González Pinto ✉ and Wolf-Tilo Balke

Institut für Informationssysteme
Technische Universität Braunschweig
Braunschweig, Germany
{pinto,balke}@ifis.cs.tu-bs.de

Abstract. This paper introduces the novel problem of 'claim-based queries' and how digital libraries can be enabled to solve it. Claim-based queries need the identification of a key aspect of research papers: claims. Today, claims are hidden in its unstructured, free text representation within research documents. In this work, a claim is a sentence that constitutes the main contribution of a paper and expresses an association between entities of particular interest in a given domain. In the following, we investigate how to identify claims for subsequent extraction in an unsupervised fashion by a novel integration of neural word embedding representations of claims with a graph based algorithm. For evaluation purposes, we focus on the medical domain: all experiments are based on a real-world corpus from PubMed, where both, limitations and success of our solution can realistically be assessed.

Keywords: claim-based queries, word embeddings, claim extraction

1 Introduction

The world is becoming an increasingly complex place, where information needs are not always simple to satisfy –even by sophisticated information retrieval algorithms over large digital libraries with carefully curated content. In this work, we introduce the novel problem of '*claim-based queries*' and show how to use focused indexing in digital libraries to reliably capture claims and subsequently answer respective queries.

So, what are claim-based queries? To get an intuition, consider the following example: a user interested in medical research may raise the general question of "which medication should be taken to alleviate a headache?" At first, the question may strike one as a bit naïve, since the answer will obviously be quite complex: there exist several medications with different pros and cons depending on the specific problem setting. Indeed, the main challenge of this example is that any 'good' answer has to deal with knowledge that is open to discussion and is highly dependent on some context missing in the question. In any case, users will need at least three steps to satisfy their query:

1. Find out what medications to alleviate a headache actually do exist (the *entity space* for possible answers),
2. Find documents, e.g. research papers, where each medication has been applied in particular problem settings (the *contextual space* for the above entities), and
3. Given all these documents, analyze them to decide which medicament fits the own particular context best (a *selection* or *ranking method*).

We see two basic requirements for any retrieval system to solve the problem. First, it needs to operationalize the notion of a claim-based query, and second, it needs high quality content as input. While the first part is indeed quite problematic, the second part may be solved by digital libraries offering high quality content, often curated by peer-review. However, a key semantic metadata element for such a system, the central claim(s) of each paper is usually not available. And this crucial step is the focus of this paper.

Previous work in the field of argumentation mining has shown the potential of algorithms to automatically identify argumentative structures such as claims from clearly structured online debate forums and from persuasive essays on various topics [1]. However, is it possible to find a solution for scientific collections, too? In this paper, we focus on the proper identification of claims in research papers. We concentrate our efforts to answer the following questions: How difficult is the task? Is the claim of a research paper usually in a single sentence or can it stretch over several sentences? Can extractors reliably identify claims?

Addressing this challenge, this work focuses on the automatic identification of claims in research papers in an unsupervised fashion. Previously, we have shown the key role that *claims* can play for Digital Libraries [2]. In particular, how they can assist peer-review to support high quality content. In this work, we introduce a novel integration of neural embedding representations of words within a technique that identifies claims in scientific articles. We test our approach on a representative corpus of PubMed articles with more than 1,000 different journals that have claims annotated.

The paper is organized as follows: section 2 provides definitions and the problem statement that we aim to solve. Section 3 reviews related work. In section 4, we first provide an analysis of the corpus used to assess the difficulty of the task. In particular, we perform an explorative analysis to answer whether the number of sentences in a claim varies, and whether specific vocabulary patterns at the beginning and ending of claims exist. Section 5 provides details on our experimental setup and discusses our findings. Finally, we draw conclusions in section 6 and point to future work.

2 Model and Problem Definition

In this section, we introduce the idea behind claim-based queries. We provide definitions and the problem statement that we aim at solving in this paper. In general, a claim-based query is a query that represents a specific and complex type of information need: a question whose answer is subject to discussion. In particular, this type of questions follows a problem-solution pair-pattern. Moreover, more than one solution exists. For example, "which medication should I take to alleviate a headache?" In this case, ‘medication’ is the *solution*, and ‘headache’ is the *problem*. Moreover, specific instances of *medication* could solve this particular problem. Each sentence where an association between a specific instance of the solution and the problem appears, is what we have called a *claim*. In this work, we argue that to answer claim-based queries, the proper identification of claims is a fundamental first step.

We will focus in the medical domain; thus, more specifically, the relationship part of the claim will be relationships in which the consumption of a product, a drug, a substance, etc., carries an effect for a given disease. We recognize that health information is a complicated process and thus, as our first attempt, we assume that the claims can be found by identifying the sentences that correspond to the main contributions of a paper. Therefore, the challenge to identify automatically this type of sentences is the focus of this paper. More formally, we are given a collection of m documents (research papers) from a digital library $D = \{d_1, \ldots, d_m\}$, where each document is represented as a sequence of sentences. Our task is then:

Problem Statement. (Claim detection in research papers). Given a collection of documents D, and a pair of entities e_1, e_2, we intend to identify automatically from each document in D, the sentence(s) $\{s_1, \ldots s_n\}$ where e_1, e_2 are related with the constraint that the sentence(s) belong to the set of the main contribution(s) of the paper. We approach the claim detection problem by breaking it down into two tasks:

1. Identification of the sentence(s) that represent the main contribution of a paper.
2. Identification of the sentence(s) of 1 where the entities e_1, e_2 are found.

To address task 1, for a given s_i in d, and for each $d \in D$, we determine whether the given sentences should be considered as the claim of d. To generate such a binary decision, we perform a claim detection process $claim(d)\ \forall d \in D$ formalized in the following expression:

$$ClaimDetection\ task: <s,d> \rightarrow \{0,1\}\ \ \forall s \in claim(d) \wedge d \in D \quad (1)$$

Task 2 is trivial once task 1 has been solved: it is only a pruning process to consider the sentence(s) where entities e_1, e_2 appear. For completeness, we summarize in Algorithm 1 how to solve the claim detection problem. However, in the following section, we describe the main contribution of this paper: step 4. In particular, we aim at performing this step in an unsupervised fashion.

Algorithm 1. Claim Detection Method.	
1.	**Input:** Document d, entities e_1, e_2
2.	**Output:** claims(s) where e_1 and e_2 occur
3.	Given d, split it in all its sentences $< s >$
4.	Decide for each sentence in $< s >$ whether the sentence is a claim $< s' >$
5.	For each sentence that is a claim $s \in s'$, consider only the sentences where e_1 and e_2 occur $< Claims_{e_1,e_2} >$.
6.	Return $< Claims_{e_1,e_2} >$

2.1 The TextRank Algorithm

In this section, we introduce the algorithm used in this work to find claims from scientific papers in an unsupervised fashion. Readers already familiar with the TextRank algorithm can skip this section. The algorithm called TextRank has its roots in the Natural Language Processing community[3]. TextRank is a graph-based ranking algorithm successfully used to extract keywords and sentences for the task of summarization. Recently, in [1] it has been shown to also have a positive effect on tasks related to argument mining. In particular, it was applied on online debating forum and persuasive essays corpora. The algorithm works as follows [3]:

1. Identify text units that best define the task at hand, and add them as vertices in the graph.
2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.
3. Iterate the graph-based ranking algorithm until convergence.
4. Sort vertices based on the final score. Use the values attached to each vertex for ranking/selection decisions.

For our task, the text units are sentences. TextRank used with sentences reduces the problem to select a 'similarity' metric between sentences that can lead to a good extraction. The original notion of similarity in TextRank is defined as the overlap between two sentences, which can be determined as the number of common words between the two sentences. Formally, given two sentences s_i and s_j, with a sentence being represented by the set of N_i words that appear in the sentence: $s_i = w_1^i, w_2^i, \dots, w_N^i$, the similarity of s_i and s_j is defined in [3] as:

$$Similarity(s_i, s_j) = \frac{|\{w_k | w_k \in s_i \,\&\, w_k \in s_j\}|}{\log(|s_i| + \log(|s_j|)} \quad (2)$$

In this work, we investigate the impact of a modified similarity measure, which incorporates neural representation of words. In particular, we use word embeddings created by word2vec from [4, 5]. Indeed, neural network based approaches [4–6] require only a large amount of unlabeled text data. The motivation of the use of this semantic embedding of words in vector spaces is twofold: it has been demonstrated that words with similar meanings are embedded nearby and also that natural word arithmetic of the vectors can be conveniently applied [5, 7]. Thus, to represent each sentence we sum the word2vec vectors of each of its words. We use this representation to compute different similarity metrics to plug into the TextRank algorithm.

3 Related Work

Our work builds on the Argumentation Mining field where researchers study the identification of argumentative structures in some given text. For instance, in [8] rhetorical roles of sentences were investigated to classify academic citations with respect to the citation effect. In particular, the idea of how a citation fits the argumentative structure. As features, they investigated the type of subject of the sentence, the citation type, the semantic class of main verb, and a list of indicator phrases that were manually evaluated. Work in [9, 10] studied persuasive essays from the discourse structure perspective. They introduced an approach to identify argumentative discourse structures. In their work, components such as claims and premises, and how they are connected with argumentative relations were studied. The researchers classified a pair of argument components as either support or non-support to identify the structure of argumentative discourse. After evaluating several classifiers, novel feature sets were proposed including structural, lexical, syntactic, and contextual features. In [11] a classification of argumentative sentences was introduced, namely four categories: none, major claim, claim, and premise. They used a supervised machine learning approach to learn these categories automatically, achieving a 0.72 macro-F1-score. In the work of [12] the idea of claim detection given a particular context was introduced. In particular, the work used annotated data from Wikipedia to assess a supervised machine learning approach. Another interesting approach was proposed by [13] where a method that used structured parsing information, detected claims without requiring contextual information. In [14] a relation-based approach was introduced for Argumentation Mining. In particular, the extraction of argumentative relations. The researchers introduced a detailed use case where pairs of sentences were annotated to focus on identifying argumentative relations.

Particularly related to our work, in [1] the TextRank algorithm was used to detect argumentative components in an online debating forum and persuasive essays. What makes different our approach is that we incorporate two key components to the algorithm: firstly, different similarity metrics and embedding representation of sentences based on word2vec. In [15], researchers elaborated on the appropriate annotation scheme for argumentation mining. In particular, they studied the educational domain using German newspaper editorials from the Web and English documents from forums, comments, and blogs. They found that the choice of the argument components depends on several different factors and structures used for expressing argumentation, thus no argumentation scheme fits all the possible applications where Argumentation Mining may play an important role. In [16], the IBM Haifa Research Group collected context-dependent claims and evidence (facts) relevant to a given topic from Wikipedia pages. The researchers classified evidence into three types: study, expert and anecdotal using manually curated data from Wikipedia.

4 Dataset

The primary focus of our experiments is to determine to which degree of success the TextRank algorithm, an unsupervised approach, can perform the task of claim detection in scientific articles. In particular, in the medical domain. To do so, we perform experiments on a PubMed corpus extracted using the following query pattern in PubMed "(help AND prevent) OR (lower AND risk) OR (increase OR increment AND risk) OR (decrease OR diminish AND risk) OR (factor AND risk) OR (associated AND risk)" as in [17]. Out of more than 1M articles retrieved, we used a sample of 10,000 that featured abstract and conclusion as metadata elements. We did so because the sentences in the conclusion metadata are considered as our ground truth. In this work, we hypothesized that the sentences in the conclusion metadata are a good indicator of the main contribution of the paper. Unfortunately, we cannot use as ground truth the Mesh terms of the documents because they are not sentences expressing the main contribution of the papers. Thus, the sentences of the abstract section and of the conclusions section constitute the set of sentences that the TextRank algorithm uses as input. Moreover, we will refer to the conclusions as the claims of the papers hereafter.

In this section, we report results of an exploratory analysis of our corpus. One particular problem that we wanted to understand is the complexity of the diversity in the content of the metadata available. Particularly, we shed light on the following questions: (1) what is the distribution of the number of sentences of a claim considering different journals? (2) What is the specific vocabulary at the beginning and ending of claims?

Let us start with our first question: whether the number of sentences containing claims differs considering different journals. Among the 1,000 different journals from our query pattern, we found that 3% of the journals use on average between 3 and 5 sentences to represent the claim of the research papers. In other words, the number of sentences used by the majority of the journals is between 1 and 3.

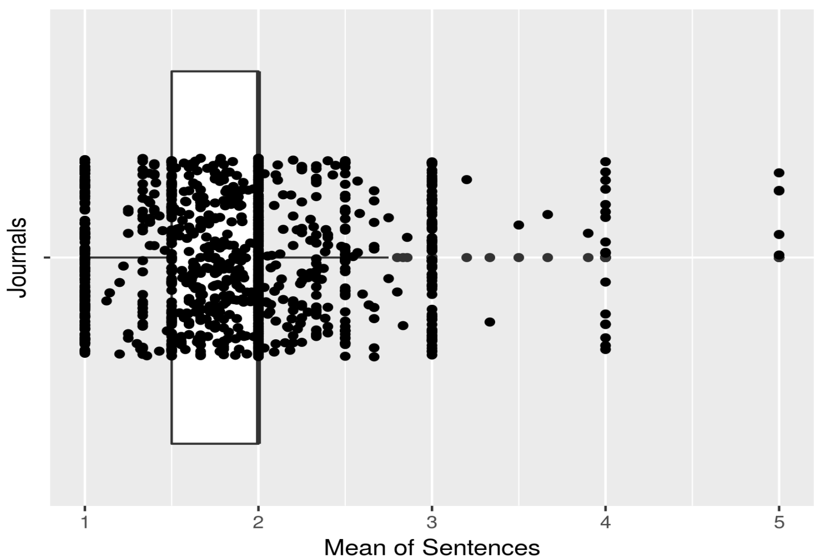

Fig. 1. Distribution of the mean number of sentences of 1K journals in PubMed

In Figure 1 we see a box plot with the mass of the mean number of sentences falling between 1 and 3 sentences. Concretely, each dot represents a different journal and the x-axis features the average number of sentences that we found in the metadata that corresponds to the claims of the papers.

Let us continue with our second question: What is the specific vocabulary at the beginning and ending of claim(s)? To answer this question, we investigated the bigrams most frequently used at the beginning and ending of the claims sections. In particular, we used the median position of the bigrams within the claims sections. In Figure 2, we plot bigrams used at least 50 times at the beginning and at the end of the claims section. It seems that there exist some text patterns than can help in the implementation of an algorithm for automatically detecting claims in medical research papers.

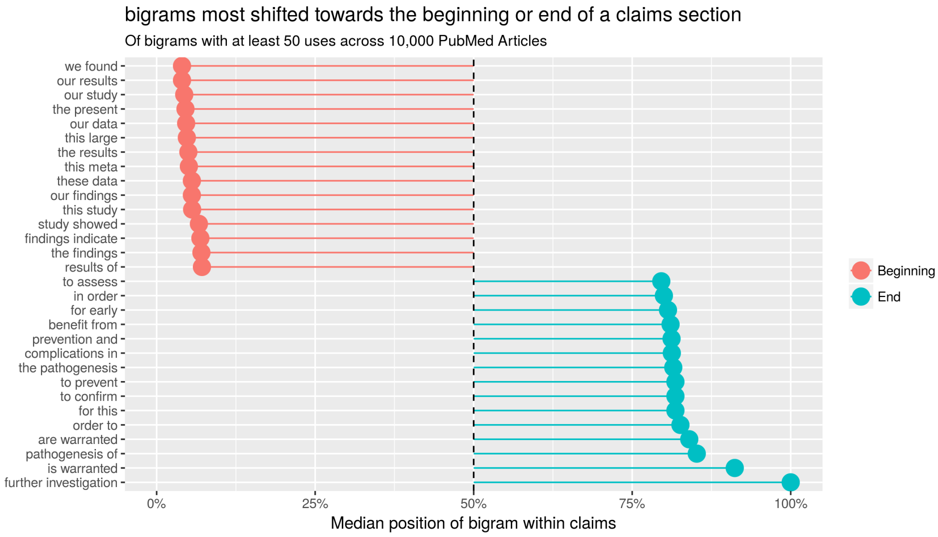

Fig. 2. Bigrams use at the beginning (left side of the graph) or end (right side of the graph) of the claim(s) section.

In Figure 2, the x-axis represents the median position of bigrams within claims. Basically, the plot divides in two main groups the bigrams of the claims sections of the papers. The first position those whose median's position are less than 50% (beginning) and the second those that are whose median's position are more than 50% (end). For instance, the bigram "is warrant" appears at the end of the claims sections, corresponding to a median position of 91.1%. Building on these insights, in the next section we proceed to provide details of the actual implementation of our approach.

5 Experiments

In this section, we report the results of our experiments. Because the number of sentences in the conclusions shows diversity (see Section 4), we also vary the number of sentences in our experiments to evaluate the performance of the implementations of

TextRank. We choose for each particular experiment different number of sentences to return considering the coverage of most of the cases we found in our exploratory analysis. Moreover, for each number of sentences we run eight different implementations of TextRank. The implementations differ in two fundamental aspects: the similarity metric used by the algorithm, and whether the implementation performs dimensionality reduction of the embedding space or not. For dimensionality reduction, we use principal component analysis (PCA) [18].

Furthermore, one of the implementations uses a Bag of Word model (BOW) with the cosine similarity as the similarity metric. We use that simple implementation to determine if the use of the word embedding for this particular task makes a difference. To compare the variations of the algorithm, we evaluate whether the returned sentence of TextRank is in the conclusions metadata. In case it is contained, we consider the sentence as correctly identified. Otherwise, it is considered incorrect. Thus, we report accuracy as the measure of success of the different algorithm's variations. In the following, we describe the variations of TextRank we evaluate.

1. BOW+TF-IDF: uses a bag of words model with TF-IDF [18] to compute cosine similarity between the sentences.
2. Embedding: uses cosine as the similarity metric. Each sentence is represented as the sum of the individual word vectors in a 200-dimensional space.
3. Embedding + Hellinger: uses the Hellinger similarity metric.
4. Embedding + PCA + Cosine: uses PCA dimensionality reduction in the word vectors. Each sentence is a sum of vectors of its individual words, but in a reduced space. Uses the Cosine similarity metric.
5. Embedding + PCA + Hellinger: similar to (4) but uses the Hellinger similarity metric.
6. Embedding + PCA + 2-Norm Diff: similar to (4) but uses the Euclidean distance of the difference of the vectors that represent each sentence as the similarity metric.
7. Embedding + PCA + 2-Norm Avg: similar to (4) but uses the Euclidean distance of the average of the vectors that represent each sentence as the similarity metric.
8. Embedding + PCA + 2-Norm Diff & Avg: similar to (4) but using the concatenation of the vectors that represent the differences and the average word2vec vectors of the sentences.

For the PCA variations, to determine the number of components to use, we use a measure known as 'explained variance', which can be calculated from the respective eigenvalues. Concretely, the explained variance tells us how much information can be attributed to each of the principal components. We experiment with different variances to empirically select the number of components and report the best results in this work.

To clarify our findings, we first provide an analysis of cases where the ground truth consists of two sentences and second, all cases where the ground truth has three sentences.

Let us start with the first case. We can observe from Table 1 that all the variations of TextRank using an embedding representation of the sentences outperform the Bag of Words model representation. This was expected, because word embedding capture semantics and syntactic features non-existent in the Bag of Words model. What is interesting to notice is that a sum over the word vectors of a sentence preserves these properties.

With respect to the similarity metric in the embedding space when PCA was not applied, the cosine similarity outperforms the Hellinger similarity with a very low margin when the top number of sentences returned by TextRank is k=3 and k=4 but the Hellinger similarity is a better choice when k=2. Thus, for our particular task of retrieving claims in an unsupervised fashion, we consider both similarity metrics equally valuable. However, when using PCA the cosine similarity has no competition. In fact, this particular implementation of the TextRank algorithm delivers the overall best results. Moreover, the Hellinger distance was consistently outperformed by the implementation that uses the norm between the average representations of the vectors as the similarity metric.

Our finding confirms the work of [7], where a similar representation of the sentences performed on par with more computationally expensive deep learning models of sentences in the task of document classification. As expected, all the implementations increase performance as we increase the number of sentences that the algorithm returns. Nevertheless, considering that the ground truth only consists of k=2 sentences, we can observe that all the implementations performed poorly on the task.

Table 1. Accuracy of the different variations of TextRank to identify claims. The value of k represents the number of sentences used to compute the accuracy

TextRank Variation	***k*=2**	***k*=3**	***k*=4**
BOW+TFIDF	0.338	0.466	0.582
Embedding	0.418	0.566	0.662
Embedding + Hellinger	0.433	0.562	0.659
Embedding + PCA + Cosine	0.463	0.609	0.701
Embedding + PCA + Hellinger	0.383	0.510	0.613
Embedding + PCA + 2-Norm Diff	0.339	0.500	0.638
Embedding + PCA + 2-Norm Avg	0.393	0.550	0.679
Embedding + PCA + 2-Norm Diff & Avg	0.378	0.535	0.662

Let us continue with the experiments that correspond to cases where the number of sentences in the ground truth is three. We present the results in Table 2. Similar to what we observe in Table 1, any embedding representation outperforms the Bag of Word model. With respect to the similarity metric when PCA was not used, we cannot see a clear winner between Hellinger and the Cosine similarity metrics. However, when we perform PCA on the word vectors, the Cosine similarity shines outperforming the Hellinger similarity metric. Nevertheless, a fundamental difference between Table 1 and Table 2, is that the method with best results in Table 2 is not the Cosine similarity with PCA but rather the implementation of the 2-Norm distance using the average vector of the sentences.

Table 2. Accuracy of the different variations of TextRank for the second test case to identify claims. The value of k represents the number of sentences used to compute the accuracy

TextRank Variation	***k*=2**	***k*=3**	***k*=4**
BOW+TFIDF	0.548	0.685	0.789
Embedding	0.652	0.775	0.858
Embedding + Hellinger	0.659	0.765	0.857
Embedding + PCA + Cosine	0.729	0.814	0.900
Embedding + PCA + Hellinger	0.631	0.723	0.821
Embedding + PCA + 2-Norm Diff	0.709	0.821	0.884
Embedding + PCA + 2-Norm Avg	0.746	0.844	0.904
Embedding + PCA + 2-Norm Diff & Avg	0.735	0.840	0.908

Discussion. In summary, we found that using an embedding representation of the sentences had a positive impact for our particular task. Furthermore, when dimensionality reduction was applied to the word vectors, with PCA, we obtained better results. Moreover, we also observed that as the parameter *k* that represents the number of sentences to extract is increased, an embedding representation with dimensionality reduction delivered the best results. In practice, we will have to make a decision regarding the number of sentences the algorithm should return. This aspect of the algorithm remains as a parameter that practitioners have to set empirically. We observed that the approach shows potential to solve the claim detection problem in the medical domain. However, more work needs to be done to improve the quality of the results. In particular, for Digital Libraries where high quality is essential, we consider that the current accuracy should be improved. And one particular way to improve the approach that we are currently considering is the use of attention mechanisms such as the one in [19]. With such an approach, the model of the sentences could be more robust to different word orders and in turn might increase the quality of the results.

6 Conclusions

In this work, we have introduced the novel problem of claim-based queries and argued how digital libraries can be enabled to solve it. One of the key parts of our solution to the problem, the automatic identification of claims in an unsupervised fashion, was in detail investigated and evaluated in this paper. In particular, the use of TextRank, a graph based algorithm, for the novel task of extracting claims of medical scientific articles. We performed a series of experiments, where we incorporated representations of sentences based on word embedding using word2vec with different similarity metrics with and without dimensionality reduction, using PCA. The representation of sentences using PCA turned out to provide best results in our evaluation with accuracy rate of over 70%. We evaluated our approach on a crawled corpus from PubMed and used all available manually assigned metadata as ground truth.

Although our results look encouraging for focused indexing of the claims found in a digital collection, in future work we need to further improve the unsupervised detection of claims. In particular, we would like to incorporate word order in the model that represent the sentences. Moreover, towards our goal of enabling digital libraries to answer claim-based queries, we would like to study the impact of claim indexing to investigate the features that can help to rank documents given a claim-based query.

7 References

1. Petasis, G., Karkaletsis, V.: Identifying Argument Components through TextRank. In: ACL. pp. 94–102 (2016).
2. González Pinto, J.M.; Balke, W.-T.: Can Plausibility Help to Support High Quality Content in Digital Libraries? In: TPDL 2017 – 21st International Conference on Theory and Practice of Digital Libraries. , Thessaloniki, Greece. (2017).
3. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. Proc. EMNLP. 85, 404–411 (2004).
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. Nips. 1–9 (2013).
5. Mikolov, T., Corrado, G., Chen, K., Dean, J.: Efficient Estimation of Word Representations in Vector Space. Proc. Int. Conf. Learn. Represent. (ICLR 2013). 1–12 (2013).
6. Collobert, R., Weston, J.: A unified architecture for natural language processing. In: Proceedings of the 25th international conference on Machine learning - ICML '08. pp. 160–167 (2008).
7. Lev, G., Klein, B., Wolf, L.: In defense of word embedding for generic text representation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 35–50 (2015).
8. Teufel, S., Phd, G.H.: Argumentative Zoning: Information Extraction from Scientific Text, (1999).
9. Stab, C., Gurevych, I.: Identifying Argumentative Discourse Structures in Persuasive Essays. Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. 46–56 (2014).
10. Stab, C., Kirschner, C., Eckle-Kohler, J., Gurevych, I.: Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In: CEUR Workshop Proceedings (2014).
11. Stab, C., Gurevych, I.: Annotating Argument Components and Relations in Persuasive Essays. Proc. COLING 2014, 25th Int. Conf. Comput. Linguist. Tech. Pap. 1501–1510 (2014).
12. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., Slonim, N.: Context Dependent Claim Detection. In: International Conference on Computational Linguistics. pp. 1489–1500 (2014).
13. Lippi, M., Torroni, P.: Context-independent claim detection for argument

mining. In: IJCAI International Joint Conference on Artificial Intelligence. pp. 185–191 (2015).
14. Carstens, L., Toni, F.: Towards relation based Argumentation Mining. Proc. 2nd Work. Argumentation Min. 29–34 (2015).
15. Habernal, I., Eckle-Kohler, J., Gurevych, I.: Argumentation Mining on the Web from Information Seeking Perspective. Proc. Work. Front. Connect. between Argumentation Theory Nat. Lang. Process. 26–39 (2014).
16. Rinott, R., Dankin, L., Alzate, C., Khapra, M.M., Aharoni, E., Slonim, N.: Show Me Your Evidence – an Automatic Method for Context Dependent Evidence Detection. Emnlp. 440–450 (2015).
17. Ciccarese, P., Wu, E., Wong, G., Ocana, M., Kinoshita, J., Ruttenberg, A., Clark, T.: The SWAN biomedical discourse ontology. J. Biomed. Inform. 41, 739–751 (2008).
18. Leskovec, J., Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press (2014).
19. Li, J., Luong, M.-T., Jurafsky, D.: A Hierarchical Neural Autoencoder for Paragraphs and Documents. 1106–1115 (2015).