Semantic Textual Similarity Measures for Case-Based Retrieval of Argument Graphs

Mirko Lenz, Stefan Ollinger, Premtim Sahitaj, and Ralph Bergmann

Business Information Systems II
University of Trier
54296 Trier, Germany
info@mirko-lenz.de, {ollinger,sahitaj,bergmann}@uni-trier.de
http://www.wi2.uni-trier.de

Abstract. Argumentation is an important sub-field of Artificial Intelligence, which involves computational methods for reasoning and decision making based on argumentative structures. This paper contributes to case-based reasoning with argument graphs in the standardized Argument Interchange Format by improving the similarity-based retrieval phase. We explore a large range of novel approaches for semantic textual similarity measures (both supervised and unsupervised) and use them in the context of a graph-based similarity measure for argument graphs. In addition, the use of an ontology-based semantic similarity measure for argumentation schemes is investigated. With a range of experiments we demonstrate the strengths and weaknesses of the various methods and show that our methods can improve over our previous work. Our code is publicly available on GitHub[1].

Keywords: Argument Graph Similarity · Semantic Textual Similarity · Argument Retrieval

1 Introduction

Argumentation is an increasingly important sub-field of Artificial Intelligence (AI). It involves various computational methods for reasoning and decision making, which are not only based on individual facts, but on coherent argumentative structures. The German special research program *Robust Argumentation Machines* (RATIO)[2] aims at developing new methods for extracting arguments from documents as well as new semantic models and ontologies for the deep representation of arguments which allows argument-based reasoning for various kinds of real-world problem solving. The major challenge is the development of so-called *argumentation machines* [27], which are specialized in reasoning with arguments. An argumentation machine could find supporting and opposing arguments for a user's topic or it could synthesize new arguments for an upcoming,

[1] https://github.com/MirkoLenz/ReCAP-Argument-Graph-Retrieval
[2] http://www.spp-ratio.de/home/

not yet well explored topic. Thereby it could support researchers, journalists, and medical practitioners in various tasks, overcoming the very limited support provided by traditional search engines used today.

In the ReCAP project [6], which is part of the RATIO program, we aim at combining methods from case-based reasoning (CBR), information retrieval (IR), and computational argumentation (CA) to contribute to the foundations of argumentation machines. In previous work [5], we developed an initial version of a similarity measure for arguments represented as argument graphs [7] for the purpose of case-based argument retrieval. This similarity measure was inspired by our own previous work on process-oriented CBR (POCBR), in which the similarity of graphs is assessed that represent semantically annotated workflows [4]. Argument graphs, however, are largely based on textual representations of claims and premises and thus require the use of textual similarity measures, thereby pushing this work closer to the sub-field of textual CBR [35]. While in our previous work, we only apply a standard word embedding technique for the assessment of local textual similarities, the aim of this paper is to explore a larger range of new approaches for semantic textual similarity measures (both supervised and unsupervised) used in the context of a graph-based similarity measure for argument graphs. In addition the use of an ontology-based semantic similarity measure for argumentation schemes is investigated.

Next, we present the foundations and related work in the field. Section 3 introduces our general approach for argument graph similarity as well as the spectrum of semantic textual similarity measures and the argumentation scheme similarity, which are the major contributions of this paper. The various methods and selected combinations of them are systematically evaluated in Section 4. Finally, Section 5 concludes the paper.

2 Foundations and Related Work

In the field of CA, an argument consists of a set of premises and a claim together with a rule of inference which concludes the claim from the premises. A premise can support or oppose a claim as well as an inference step. Together premises, claims, and inference steps form an argument graph. The Argument Interchange Format (AIF) standardizes such a graph representation for arguments [15] to be used in CA. In Fig. 1 an example of an argument graph in AIF format is given. Claims and premises are represented as information nodes (I-nodes), depicted as rectangular boxes which are related to each other via scheme nodes (S-nodes), depicted as rhombuses. In the example there are two arguments for a claim related to health insurance. The opposing argument has a single premise, whereas the supporting argument has two distinct premises. Argumentation schemes, corresponding to archetypical forms of arguments, are annotated as types of an argument. Here, the supporting argument has a type of *Position to Know*. The opposing argument has the type *Default Conflict*. There are many different argumentation schemes which cover diverse facets of argumentation [34], such

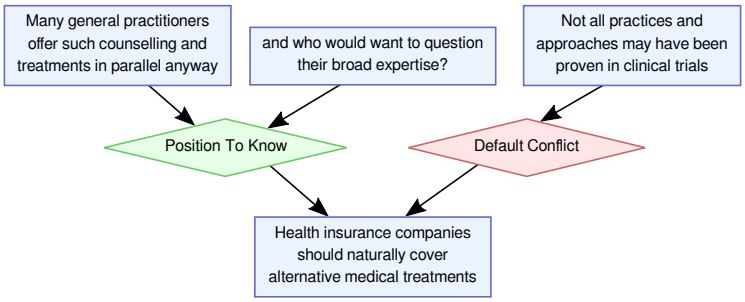

Fig. 1. Example of an argument graph in AIF format from the Microtexts corpus [25]

as *Argument from Positive Consequence*, *Argument from Expert Opinion*, or *Argument from Cause to Effect*.

While argument mining methods [22] aim at converting natural language argumentative texts into such argument graphs, our work aims at supporting the reasoning with such graphs. Several formal argumentation frameworks currently exist which are based on formal logic, but we believe that they are of limited use for future argumentation machines that reason with real-world arguments [12]. Thus, we propose CBR as it does not require a complete and consistent domain theory and is able to make use of vague information. Thereby, we continue the traditional path for the use of textual CBR [35] in the context of argumentation for legal reasoning [3,11] and aim at linking it with ideas from POCBR and novel semantic text similarity approaches.

Existing work on CBR for legal argumentation is based on a model of legal argument. Cases are represented based on hierarchically structured factors or issues [29], which are used during similarity-based retrieval. A factor is similar to an argument or premise. The similarity of two arguments is defined by the commonalities and differences of the factors. CATO extends those argument graphs with intermediate factors, forming a factor hierarchy [1]. Branting [11] proposes case-based adaptation in legal reasoning by reusing and adapting justifications to create new arguments. Interestingly, similar ideas have been recently established in the field of CA such as the "recycling" of arguments for synthesis of claims [8].

3 Argument Graph Retrieval using Semantic Textual Similarities

We now describe our approach to the representation of cases in the form of semantically labeled argument graphs, we recapitulate the basic approach for similarity assessment [5], and introduce the main enhancements by semantic textual similarity measures and the argumentation scheme ontology.

3.1 Argument Graph Representation

We developed a case representation using argument graphs, which is based on the graph representation of AIF. It is similar to text reasoning graphs [31] for representing causal information, but argument graphs contain in addition semantic information in different forms. Formally, an argument graph is a semantically labeled directed graph and represented as a tuple $A = (N, E, \tau, \lambda, t)$ [4]. N is the set of nodes and $E \subseteq N \times N$ is the set of directed edges connecting two nodes. $\tau : N \to \mathcal{T}$ assigns each node a type and $\lambda : N \to \mathcal{L}$ assigns each node a semantic description from a language $\mathcal{L}$. $t \in \mathcal{L}$ describes the overall topic of the argument represented in the graph. The types $\mathcal{T}$ follow the AIF standard [15] so that a node can either be an I-node with natural language propositional content or an S-node characterized by the respective argumentation scheme. The mapping function λ is used to link a semantic representation to a node. For an I-node n, $\lambda(n)$ is the original textual representation (possibly after the application of traditional pre-processing such as stopword removal) together with a semantic representation of this text in the form of a vector, produced by a sentence encoder (see Sec. 3.3). For an S-node n, $\lambda(n)$ corresponds to an argumentation scheme identifier, from an ontology of argumentation schemes constructed following the classification as proposed by Walton [33]. The argumentation scheme ontology is further used to define a local similarity measure for comparing two S-nodes, as described in Sec. 3.4. Finally, the overall topic t of an argument graph corresponds to the concatenated textual contents of all I-nodes as well as their semantic vector representation.

For retrieval, a case base of argument graphs is assumed, which could result from argument mining or from the manual transformation of text corpora. In our work, we also consider a query to be an argument graph or a fraction of it. In particular, a query can also consist only of a single I-node.

3.2 Argument Graph Similarity and Retrieval

The general principle of argument graph similarity and retrieval introduced by Bergmann et al. [5] has been adopted from POCBR [4] and follows the local-global principle [28]. The global similarity is computed from local node and edge similarities. The local node similarity $\mathrm{sim}_N(n_q, n_c)$ of a node n_q from the query argument graph QA and a node n_c from the case argument graph CA is computed as follows:

$$\mathrm{sim}_N(n_q, n_c) = \begin{cases} \mathrm{sim}_I(n_q, n_c), & \text{if } \tau(n_q) = \tau(n_c) = \text{I-node} \\ \mathrm{sim}_S(n_q, n_c), & \text{if } \tau(n_q) = \tau(n_c) = \text{S-node} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Approaches for concrete I-node and S-node similarity functions sim_I and sim_S are the main contribution of this paper and introduced in the next subsections. The similarity of two edges $\mathrm{sim}_E(e_q, e_c)$ is the average of the similarities of their endpoints l and r respectively:

$$\mathrm{sim}_E(e_q, e_c) = 0.5 \cdot (\mathrm{sim}_N(e_q.l, e_c.l) + \mathrm{sim}_N(e_q.r, e_c.r)) \tag{2}$$

To construct a global similarity value, an admissible mapping m is applied which maps nodes and edges from QA to CA, such that only nodes of the same type (I-nodes to I-nodes and S-nodes to S-nodes) are mapped. Edges can only be mapped if the nodes they link are mapped as well by m. For a given mapping m let sn_i be the node similarities $\text{sim}_N(n_i, m(n_i))$ and se_i the edge similarities $\text{sim}_E(e_i, m(e_i))$. The similarity for a query graph QA and a case graph CA given a mapping m is the normalized sum of the node and edge similarities.

$$\text{sim}_m(QA, CA) = \frac{sn_1 + \cdots + sn_n + se_1 + \cdots + se_m}{n_N + n_E} \tag{3}$$

Finally, the similarity of QA and CA is the similarity of an optimal mapping m, which can be computed using an A^* search [4].

$$\text{sim}(QA, CA) = \max_m \{\text{sim}_m(QA, CA) \mid m \text{ is admissible}\} \tag{4}$$

For similarity-based retrieval of argument graphs from a case base a linear retrieval approach should be avoided due to unacceptable retrieval times caused by the complexity of A^* search as well as the complexity of the involved node similarity measures. Thus, we propose a MAC/FAC *(many are called, but few are chosen)* approach [17], which divides the retrieval into an efficient pre-filter stage (MAC phase) and the subsequent FAC phase, in which only the a few filtered cases are assessed using the complex similarity measure. We proposed a MAC/FAC approach for argument graphs in which the MAC phase is implemented as a linear similarity-based retrieval of the cases based only on the semantic similarity of the topic vector t [5]. The filter selects the k most similar cases, which are passed over to the FAC phase which implements the ranking by a linear assessment of the cases using the graph-based similarity as described above.

3.3 Semantic Textual Similarity Measures for I-Node Similarity

The quality of the overall similarity assessment heavily depends on the applied node similarity measures. In our previous work we only employed Word2vec Skip-gram [23] embeddings aggregated with an arithmetic mean and compared with a cosine similarity. In this paper we investigate a larger, more diverse set of novel methods for semantic textual similarity based on neural networks. The approaches include unsupervised word and sentence embeddings and their combination as well as supervised sentence embeddings which are trained on a large amount of training data. There are also other methods available like SIF [2] or Skip-Thought vectors [19] which however will not be evaluated here.

Unsupervised Word Embeddings Word embeddings are distributed representations of words, which means each word is associated with a word vector. Word vectors capture the semantics of a word, in the sense that similar words have similar word vectors. Word embedding models are trained on textual data

in an unsupervised manner. The models rely on the distributional hypothesis, namely that words in similar contexts share meaning.

Word2vec Skip-gram (WV) [23] trains word vectors based on the prediction of context words. The model architecture employs a softmax classificator and maximizes the log likelihood of the word vectors based on (word, context) pairs. Words appearing in similar contexts have therefore similar word vectors. For performance reasons the softmax is replaced by either a hierarchical softmax or an alternative negative sampling objective [24]. The fastText (FT) embedding [9] is based on the Skip-gram model. In addition it uses subword information as each word is represented as the sum of its character n-grams together with the word. A vector for n-grams is learned which allows to build word representations for previously unseen words. GloVe (GL) [26] learns word vectors from global corpus statistics directly, in contrast to Skip-gram's context window approach. An objective function based on ratios of co-occurrence probabilities is maximized.

In order to assess the similarity of I-nodes, the individual embeddings of the words in the node's text have to be aggregated to an overall node embedding, based on which the similarity can be assessed, e.g. by a cosine measure. Traditional unsupervised aggregation methods for this task include arithmetic mean ($\overline{x}_a$), median ($\overline{x}_m$), geometric mean ($\overline{x}_g$), min pooling ($\min x$), max pooling ($\max x$) and p-means ($\overline{x}_p$) [30].

Unsupervised Sentence Embeddings Sentence embedding methods are an alternative approach that can be applied to assess the similarity of the I-nodes based on their text. As they work on sentences rather than on words, no aggregation is needed. The Distributed Memory Model of Paragraph Vectors (DV) [21] is such a method trained similarly to word2vec's CBOW model [23], but with an additional vector representing the sentence as a whole. Embeddings for previously unseen sentences are inferred by backpropagation on the paragraph vector keeping all other parameters fixed.

Supervised Sentence Embeddings While the previous embedding approaches are purely unsupervised, several approaches exist which aim at improving the similarity assessment including to a certain degree also supervised learning, thereby accepting the additional effort caused by labeled training data. InferSent [16] is one such approach trained on the Stanford Natural Language Inference corpus [10]. During training a shared BiLSTM encodes two sentences and the encoded sentence pair is further enhanced with additional features, such as the absolute difference of both sentences and their element-wise product, before it is passed through a feed-forward network for classification. After training the BiLSTM yields a 4096 dimensional vector for a sentence. Universal Sentence Encoder [13] trains a sentence encoder on multiple unsupervised and supervised tasks. The transformer-based variant (USE-T) uses a transformer encoder [32]. Deep Average Network-based Universal Sentence Encoder (USE-D) uses a Deep Average Network encoder [18] instead, which averages unigram- and bigram-

embeddings and passes the averaged value through a feed-forward network. The output of both networks is a 512 dimensional vector, representing a sentence.

Combining Different Embeddings The various embeddings just described can also be combined, following the idea that each embedding type captures different kinds of information [30]. The concatenation of two embeddings A and B is denotated by A $\oplus$ B, resulting in an embedding with dimension dim(A) + dim(B). For example WV $\oplus$ FT is the concatenation of WV and FT embeddings.

Similarity Measures for I-Nodes In order to assess the similarity of I-nodes, a similarity measure is required which compares the computed embedding vectors of the nodes. Traditionally, the cosine similarity is used in semantic textual similarity tasks, but various alternative approaches exist. The MaxPool-Jaccard approach applies the fuzzy Jaccard index to max pooled word embeddings and has recently demonstrated a significant benefit in semantic textual similarity tasks [37]. In addition, the DynaMax-Jaccard approach was proposed, which is a completely unsupervised and non-parametric similarity measure that dynamically extracts and max-pools good features.

Finally, I-node similarity can be computed using the Word Mover's Distance (WMD) [20] which computes the distance of two sentences by a mapping between the word embeddings of the sentences. An optimal mapping is found by taking into account the distances of the words in a word embedding space, so that each word in one sentence needs to travel the lowest distance to the words in the other sentence.

Please note that WMD, DynaMax and MaxPool do not operate on the representation of the whole node text but on the representation of the individual words. As such they combine aggregation and similarity assessment.

3.4 Ontology-based Similarity Measure for S-Node Similarity

We now introduce an approach with which we aim to improve the similarity assessment of argument graphs by considering the semantics of the argumentation schemes used in the S-nodes of the graph. In our previous work [5] we only used two different schemes and an exact match similarity. We now introduce a more fine grained representation and created an ontology consisting of 38 argumentation schemes which are arranged in a taxonomy based on a classification of argumentation schemes [33]. Fig. 2 shows an excerpt of this ontology.

The similarity between two schemes can then be computed by using an edge-count based approach. Wu and Palmer introduce a similarity measure sim_{wp} that considers the depth of schemes S_1, S_2 and the closest common predecessor scheme S_x of S_1 and S_2. The Wu and Palmer similarity of two argumentation schemes S_1 and S_2 [36] is given by

$$\text{sim}_{wp}(S_1, S_2) = \frac{2N_x}{N_1 + N_2} \tag{5}$$

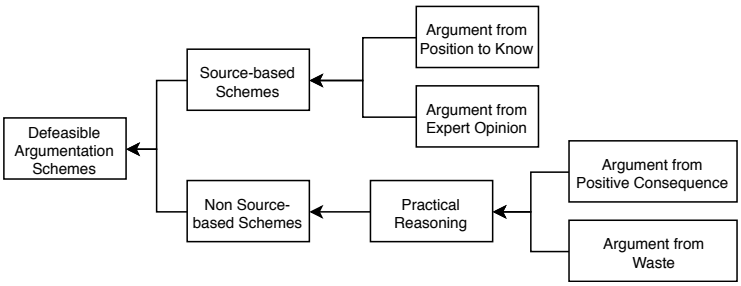

Fig. 2. Excerpt from the argumentation scheme ontology based on a classification of argumentation schemes [33]

where N_1, N_2 and N_x describe the depth of the schemes S_1, S_2 and S_x respectively in terms of edges from root element to scheme. Wu and Palmer similarity assumes that schemes located deeper in the ontology are more specific and therefore more similar.

4 Experimental Evaluation

Given the various approaches described so far for I-node similarity as well as the advanced approach for S-node similarity, we now want to experimentally evaluate the benefit of them. Thus, we performed a systematic evaluation to test how well the various approaches are able to retrieve and rank cases in a way that is in line with the assessment of a human expert.

4.1 Hypotheses

The following four hypotheses are subject of this evaluation and relate to the quality of the ranking produced by the argument graph similarity.

- **H1:** The simple approach based on WV embeddings, mean aggregation, and cosine similarity as investigated in previous work [5] can be improved by some of the newly investigated methods.
- **H2:** The concatenation of embeddings achieves a higher quality than a single embedding.
- **H3:** Supervised sentence embeddings achieve a higher quality than unsupervised embeddings.
- **H4:** The use of argumentation scheme similarity improves the quality.

4.2 Experimental Setup

For the evaluation we rely on various pre-trained word embeddings and sentence encoder models. Word2vec GoogleNews[3] vectors are trained on the Google News

[3] https://code.google.com/archive/p/word2vec/

dataset on about 100B tokens. GloVe[4] is trained on the Common Crawl dataset on 840B tokens. fastText[5] vectors are trained on Wikipedia and Common Crawl. The dimensionality of all word embeddings is 300. The Paragraph Vector model was trained by us on the english Wikipedia dump with 1M tokens. The Universal Sentence Encoder models[6,7] are trained on multiple unsupervised and supervised tasks, such as predicting context sentences [19] and classification on the SNLI corpus. InferSent[8] is trained on the SNLI corpus as well. We evaluate the model in version 1.

For the evaluation of the retrieval we choose the annotated corpus of argumentative microtexts [25] following the work in [5]. This corpus consists of 112 argument graphs with a total of 576 I-nodes and 443 S-nodes. For this paper, we refined these argument graphs by introducing a more fine-grained annotation of the S-nodes by argumentation schemes based on the ontology developed. These refinements were made by two students who were experienced in AIF and the OVA+ modelling tool[9]. For our evaluation, we used the same 24 queries from 6 topics as in our previous work. However, due to the introduction of the more detailed argumentation scheme representation, a new reference ranking was needed. It was produced by the same experienced students who refined the representation of the cases.

In our experiments, we used various combinations of the similarity methods proposed for retrieval of cases. In each experiment all 24 queries are used and the resulting k=10 most similar cases are considered. We assessed the relevance of the cases (i.e. whether a case deals with the same topic as the query) as well as the ranking of the cases. Thereby, the similarity measures are evaluated by means of various metrics. Precision (P) measures the fraction of relevant cases retrieved within the set of 10 most similar cases. Due to the size of the reference rankings in our experiment, the upper limit for P achievable is 0.717. Recall (R) measures the fraction of relevant cases retrieved. P and R are set-based,i.e., the ranking quality itself is not assessed.

Average Precision (AP) measures the quality of the ranking by averaging the precision at all relevant positions. Thus, AP is the area under the precision-recall curve. A high AP value indicates that relevant elements are ranked high as well.

Normalized Discounted Cumulative Gain (NDCG) assesses that elements with a high relevance appear early in the ranking. NDCG is computed as the normalized sum of all relevance values in the result giving lower positions in the ranking less weight. It is noteworthy that for NDCG non-relevant elements in the ranking have no influence on the metric.

Correctness (CR) and Completeness (CP) [14] explicitly assess how well the ordering of the ranking produced by similarity matches the ordering of the refer-

[4] https://nlp.stanford.edu/projects/glove/
[5] https://fasttext.cc/
[6] https://tfhub.dev/google/universal-sentence-encoder-large/3
[7] https://tfhub.dev/google/universal-sentence-encoder/2
[8] https://github.com/facebookresearch/InferSent
[9] http://ova.arg-tech.org/

ence ranking. CP measures the percentage of pairs of the reference ranking that are included the produced ranking. CR measures concordance/disconcordance of the orderings of those pairs, resulting in a value from $[-1, 1]$ with higher values indicating higher concordance. It is important to note that CR values are only meaningful if also the CP value is high. Thus we only interpret CR values if the CP value is above 0.9.

In the following we always report values averaged over all queries. In addition, we show the average retrieval time in seconds on a 2014 MacBook Pro 15" with a 2.8 GHz Intel Core i7 processor and 16 GB RAM.

4.3 Results and Discussion

In the following experiments the similarity measures for I-nodes are evaluated. Only stopword removal is consistently performed in all conditions as this was the most successful pre-processing approach in our previous work. The S-node similarity measure is evaluated lastly.

The first experiment evaluates WV embeddings together with the cosine measure as in our previous work, but using various aggregation functions. The results are shown in Tab. 1, while the abbreviations are used as introduced in Sections 3.3 and 4.2.

Arithmetic mean performs best regarding the unranked measures P and R and also w.r.t. AP. For the ranked measures, max pooling led to the best NDCG value, but for a significantly lower recall. Median achieves best results for the ranked measure correctness among all aggregations with a completeness above 0.9. We systematically evaluated also concatenations of aggregation functions without being able to outperform the individual methods. The two best concatenation results are shown in the last two rows of Tab. 1.

Next we evaluate all unsupervised and supervised embedding methods using cosine similarity. Word embeddings are aggregated using arithmetic mean. The

Table 1. Results of retrieval with WV and cosine using different aggregation functions.

Aggregation Method	Time	P	R	AP	NDCG	CR	CP
$\overline{x}_a$	10.625	**0.692**	**0.965**	**0.924**	0.834	0.106	**0.956**
$\overline{x}_m$	11.021	0.675	0.943	0.903	0.844	**0.139**	0.907
$\overline{x}_g$	6.311	0.017	0.021	0.003	0.053	–	0.0
$\min x$	10.515	0.604	0.84	0.798	0.856	0.171	0.744
$\max x$	12.687	0.588	0.827	0.786	**0.866**	0.127	0.696
$\overline{x}_2$	9.663	0.65	0.908	0.836	0.825	0.14	0.846
$\overline{x}_3$	8.059	0.654	0.913	0.876	0.849	0.146	0.876
$\overline{x}_5$	9.932	0.608	0.853	0.805	0.84	0.064	0.734
$\overline{x}_{10}$	8.831	0.575	0.81	0.746	0.843	0.174	0.676
$\overline{x}_{1000}$	6.969	0.479	0.667	0.584	0.802	0.184	0.471
$\overline{x}_a \oplus \overline{x}_m$	11.676	**0.692**	**0.965**	0.923	0.835	0.115	0.948
$\overline{x}_a \oplus \overline{x}_m \oplus \overline{x}_2 \oplus \overline{x}_3$	11.52	**0.692**	**0.965**	0.918	0.841	0.116	0.952

concatenation of the unsupervised embeddings have also been evaluated systematically, while only the best results are reported. Table 2 shows the results.

Table 2. Results of retrieval with different embedding methods using cosine and arithmetic mean.

Embedding Method	Time	P	R	AP	NDCG	CR	CP
DV	12.132	0.675	0.942	0.888	0.854	0.148	0.9
FT	11.312	0.675	0.943	0.909	0.847	0.141	0.914
GL	12.201	0.667	0.929	0.876	0.809	0.044	0.897
WV	10.625	0.692	0.965	0.924	0.834	0.106	0.956
DV $\oplus$ WV	11.737	0.696	0.97	0.934	0.855	0.097	0.958
DV $\oplus$ FT $\oplus$ WV	12.489	0.671	0.938	0.905	0.846	0.085	0.904
InferSent	40.125	0.683	0.95	0.915	**0.864**	**0.184**	0.908
USE-D	8.92	0.704	0.982	0.951	0.841	0.099	0.977
USE-T	13.785	**0.713**	**0.994**	**0.972**	0.848	0.12	**0.992**

All methods achieve a high recall and completeness. Among the unsupervised methods WV achieves the best results w.r.t. the unranked measures. DV and concatenations including DV achieve the best ranked results NDCG and CR. The supervised methods further improve the results. USE-D and USE-T yield the highest P, R, and AP scores. InferSent was best w.r.t. the ranked measures, but was even worse than WV concerning P and R. This indicates that supervised methods can actually learn useful signals for semantic textual similarity. Hypothesis H3 can thus be accepted.

The impact of the similarity measure on the retrieval quality was evaluated next. WV with arithmetic mean is used as sentence embeddings. Table 3 shows the results for the different similarity measures.

Table 3. Results of retrieval with WV while using different similarity measures.

Similarity Method	Time	P	R	AP	NDCG	CR	CP
Cosine	10.625	**0.692**	**0.965**	0.924	0.834	0.106	**0.956**
DynaMax-Jaccard	12.276	**0.692**	0.964	**0.934**	**0.877**	**0.274**	0.936
MaxPool-Jaccard	9.725	0.417	0.58	0.548	0.846	*0.34*	*0.365*
WMD	88.377	0.683	0.953	0.913	0.859	0.226	0.932

Cosine and DynaMax-Jaccard perform comparably well only on the unranked measures, while DynaMax-Jaccard significantly improves the ranking results NDCG and CR compared to cosine. WMD achieves nearly comparable results, but leads to very high retrieval times and is thus not competitive. MaxPool-Jaccard is very poor on R and CP and thus not useful.

Since the DynaMax-Jaccard similarity performed best we evaluated the various unsupervised embeddings methods and their combinations again. The supervised methods could not be evaluated here, since DynaMax-Jaccard works

Table 4. Results of retrieval with different embedding methods using DynaMax-Jaccard.

Method	Time	P	R	AP	NDCG	CR	CP
DV	11.241	0.633	0.888	0.842	0.867	0.286	0.767
FT	10.497	**0.696**	**0.971**	0.93	0.868	0.256	**0.956**
GL	12.664	0.688	0.959	0.917	0.868	0.217	0.918
WV	12.276	0.692	0.964	**0.934**	0.877	0.274	0.936
DV ⊕ FT	13.236	0.688	0.959	0.914	0.869	0.277	0.918
DV ⊕ GL	14.077	0.667	0.931	0.891	**0.883**	0.274	0.812
DV ⊕ WV	13.58	0.667	0.929	0.893	0.862	0.258	0.85
FT ⊕ GL	16.772	0.675	0.943	0.907	0.872	0.278	0.899
FT ⊕ WV	12.17	**0.696**	0.970	0.924	0.862	0.27	0.943
GL ⊕ WV	15.25	0.679	0.949	0.903	0.867	0.192	0.853
DV ⊕ FT ⊕ GL	17.902	0.663	0.926	0.886	0.881	0.328	0.815
DV ⊕ FT ⊕ WV	15.251	0.692	0.964	0.922	0.875	**0.307**	0.943
DV ⊕ GL ⊕ WV	15.55	0.679	0.947	0.905	0.877	0.304	0.872
FT ⊕ GL ⊕ WV	16.721	0.671	0.938	0.897	0.873	0.264	0.832
DV ⊕ FT ⊕ GL ⊕ WV	18.551	0.679	0.948	0.908	0.868	0.222	0.888

on word embeddings and supervised methods yield sentence embeddings. The results are shown in Table 4.

Interestingly FT embeddings perform now best w.r.t. the unranked measures and are also very high in AP. Overall concatenations are able to improve the ranking quality. DV ⊕ FT ⊕ WV yields the strongest CR and very high NDCG and can even improve over the results of the supervised methods using the cosine measure. Therefore hypothesis H2 can be accepted at least for DynaMax-Jaccard. It is noteworthy that all metrics show slightly higher values than for cosine, especially correctness and NDCG (compare Tables 2 and 4). This indicates that DynaMax-Jaccard generally leads to an improved ranking.

The use of argumentation schemes for retrieval is evaluated next. Supervised embeddings are compared using cosine, unsupervised embeddings using DynaMax-Jaccard. Concerning S-node similarity, three variants are included: no S-node similarity (always 1), exact match similarity using the argumentation scheme labels at the S-nodes and the ontology similarity (see Sec. 3.4). Table 5 presents the results. Since the argumentation schemes are used only in the FAC phase only the ranked metrics are affected and reported.

For USE-T, the use of the argumentation scheme labels slightly improves the ranking CR. The ontology-based similarity measures does not lead to an improvement for any embedding, it even worsens the ranking results. Thus, hypothesis H4 has to be rejected.

To come to a concluding assessment of hypothesis H1, we compare the three best methods, USE-T, WV, and DV ⊕ FT ⊕ WV against the approach in [5] (see Tab. 6). Again the supervised embedding is compared using cosine similarity and unsupervised embeddings using DynaMax-Jaccard. Argumentation schemes are not used.

Table 5. Results of retrieval with including argumentation scheme similarity and selected embedding methods

Embedding	Schemes	Time	AP	NDCG	CR	CP
USE-T	No	13.785	**0.972**	**0.848**	0.12	0.992
USE-T	Exact Match	15.541	0.925	0.843	**0.136**	0.992
USE-T	Onto. Sim.	14.127	0.938	0.847	0.132	0.992
WV	No	12.276	**0.934**	**0.877**	**0.274**	0.936
WV	Exact Match	13.659	0.906	0.853	0.161	0.936
WV	Onto. Sim.	10.677	0.902	0.851	0.174	0.936
DV $\oplus$ FT $\oplus$ WV	No	15.251	**0.922**	**0.875**	**0.307**	0.943
DV $\oplus$ FT $\oplus$ WV	Exact Match	19.83	0.908	0.859	0.252	0.943
DV $\oplus$ FT $\oplus$ WV	Onto. Sim.	27.096	0.905	0.861	0.216	0.943

Table 6. Evaluation of the approach used in [5] compared to the best new methods.

Method	Time	P	R	AP	NDCG	CR	CP
Paper [5]	10.625	0.692	0.965	0.924	0.834	0.106	0.956
USE-T	13.785	**0.713**	**0.994**	**0.972**	0.848	0.12	**0.992**
WV	12.276	0.692	0.964	0.934	**0.877**	0.274	0.936
WV $\oplus$ FT $\oplus$ DV	15.251	0.692	0.964	0.922	0.875	**0.307**	0.943

All three methods clearly improve on the baseline. USE-T has best P, R, AP as well as CP and can reach also near the precision limit of 0.717. WV achieves very good results with minimal complexity. DV $\oplus$ FT $\oplus$ WV achieves the best CR score. Hypothesis H1 can thus be clearly accepted. Concerning the retrieval time, the new best methods are clearly more time consuming (up to 50 %), but we consider this as acceptable given the resulting quality improvements.

5 Conclusion and Future Work

In this work we investigated new methods from semantic textual similarity for improved case-based argument retrieval and demonstrated significant improvements over our own previous results [5]. Unsupervised word embeddings and concatenations achieve a good ranking quality using the DynaMax-Jaccard similarity measure and can improve clearly on the cosine similarity measure. Supervised methods achieve the best results using the unranked metrics and the highest completeness measures. The similarity measures for argumentation schemes cannot further improve these results. A possible reason could be that the use of schemes yields in too many constraints when performing the graph mapping and thus impairing the results.

In future work we want to improve the ranking quality of supervised methods as well as explore more advanced ontological similarity measures by automatic linking with domain specific ontologies. Another line of work would be to extend the argument retrieval task to new benchmark corpora and in particular corpora in German language. A big challenge is addressing semantic similarity for the

German language as most recent methods have been mainly investigated and optimized for the English language. Additionally, we will look at reducing the computational complexity of the mapping algorithm, especially the A* search. Finally, we intend to move further on to the adaptation of argument graphs by transferring compositional adaptation methods from POCBR.

Acknowledgments. This work was funded by the German Research Foundation (DFG), project 375342983.

References

1. Aleven, V.: Teaching Case-Based Argumentation Through a Model and Examples. Ph.D. thesis, University of Pittsburgh (1997)
2. Arora, S., Liang, Y., Ma, T.: A simple but though Baseline for Sentence Embeddings (2017)
3. Ashley, K.D.: Modelling Legal Argument: Reasoning with Cases and Hypotheticals. Ph.D. thesis, University of Massachusetts (1988)
4. Bergmann, R., Gil, Y.: Similarity Assessment and Efficient Retrieval of Semantic Workflows. Information Systems **40**, 115–127 (2014). https://doi.org/10.1016/j.is.2012.07.005
5. Bergmann, R., Lenz, M., Ollinger, S., Pfister, M.: Similarity Measures for Case-Based Retrieval of Natural Language Argument Graphs in Argumentation Machines. In: Proceedings of the 32nd Intert. Florida Art. Int. Research Society Conf., FLAIRS 2019, Sarasota, Florida, USA. AAAI-Press (2019)
6. Bergmann, R., Schenkel, R., Dumani, L., Ollinger, S.: ReCAP - Information Retrieval and Case-Based Reasoning for Robust Deliberation and Synthesis of Arguments in the Political Discourse. In: Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", LWDA. vol. 2191. CEUR-WS.org (2018)
7. Bex, F., Prakken, H., Reed, C.: A Formal Analysis of the AIF in Terms of the ASPIC Framework. In: Proceedings of COMMA. pp. 99–110. IOS Press (2010)
8. Bilu, Y., Slonim, N.: Claim Synthesis via Predicate Recycling. In: Proc. 54th Annual Meeting of the Association for Computational Linguistics (ACL) (2016)
9. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information (2016), https://arxiv.org/abs/1607.04606
10. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference (2015), https://arxiv.org/abs/1508.05326
11. Branting, K.: A reduction-graph model of precedent in legal analysis. Artificial Intelligence **150**(1), 59–95 (2003)
12. Caminada, M., Wu, Y.: On the Limitations of Abstract Argumentation. In: Proceedings of the 23rd Benelux Conference on Artificial Intelligence (2011)
13. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.H., Strope, B., Kurzweil, R.: Universal Sentence Encoder (2018), http://arxiv.org/abs/1803.11175
14. Cheng, W., Rademaker, M., De Baets, B., Hüllermeier, E.: Predicting partial orders: ranking with abstention. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 215–230. Springer (2010)
15. Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., Willmott, S.: Towards an argument interchange format. Knowl. Eng. Rev. **21**(4), 293–316 (2006)

16. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised Learning of Universal Sentence Representations from Natural Language Inference Data (2017), https://arxiv.org/abs/1705.02364
17. Forbus, K.D., Gentner, D., Law, K.: MAC/FAC - A model of similarity-based retrieval. Cognitive Science **19**(2), 141–205 (1995)
18. Iyyer, M., Manjunatha, V., Boyd-Graber, J., Daumé III, H.: Deep unordered composition rivals syntactic methods for text classification (2015). https://doi.org/10.3115/v1/P15-1162
19. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Torralba, A., Urtasun, R., Fidler, S.: Skip-Thought Vectors (2015), https://arxiv.org/abs/1506.06726
20. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From Word Embeddings to Document Distances. In: Proceedings of the 32nd ICML. vol. 37, pp. 957–966. JMLR.org (2015)
21. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. In: Proceedings of the 31st ICML. vol. 32, pp. II–1188–II–1196. JMLR.org (2014)
22. Lippi, M., Torroni, P.: Argument mining from speech: Detecting claims in political debates. In: Proc. 13th AAAI Conf on Artificial Intelligence. AAAI Press (2016)
23. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013), https://arxiv.org/abs/1301.3781
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality (2013), https://arxiv.org/abs/1310.4546
25. Peldszus, A., Stede, M.: An Annotated Corpus of Argumentative Microtexts. In: First European Conference on Argumentation, Portugal, Lisbon (2015)
26. Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation. In: Proceedings of EMNLP (2014). https://doi.org/10.3115/v1/D14-1162
27. Reed, C., Norman, T.J. (eds.): Argumentation Machines, New Frontiers in Argument and Computation, Argumentation Library, vol. 9. Springer (2004)
28. Richter, M.M., Weber, R.O.: Case-Based Reasoning - A Textbook. Springer (2013)
29. Rissland, E.L., Ashley, K.D., Branting, K.: Case-based reasoning and law. The Knowledge Engineering Review **20**(3), 293–298 (2005)
30. Rücklé, A., Eger, S., Peyrard, M., Gurevych, I.: Concatenated p-mean Word Embeddings as Universal Cross-Lingual Sentence Representations (2018), https://arxiv.org/abs/1803.01400
31. Sizov, G., Öztürk, P., Štyrák, J.: Acquisition and Reuse of Reasoning Knowledge from Textual Cases for Automated Analysis. In: Lamontagne, L., Plaza, E. (eds.) Case-Based Reasoning Research and Development. pp. 465–479. Springer (2014)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30, pp. 5998–6008 (2017)
33. Walton, D., Macagno, F.: A classification system for argumentation schemes. Argument and Computation **6**(3), 219–245 (2015)
34. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press (2008)
35. Weber, R.O., Ashley, K.D., Brüninghaus, S.: Textual case-based reasoning. The Knowledge Engineering Review **20**(03), 255–260 (2005)
36. Wu, Z., Palmer, M.: Verbs Semantics and Lexical Selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (1994)
37. Zhelezniak, V., Savkov, A., Shen, A., Moramarco, F., Flann, J., Hammerla, N.Y.: Don't settle for average go for the Max: Fuzzy Sets and Max-pooled Word Vectors. In: International Conference on Learning Representations (2019)