Pairwise Multi-Class Document Classification for Semantic Relations between Wikipedia Articles

Malte Ostendorff[1,2],Terry Ruas[3], Moritz Schubotz[3], Georg Rehm[1], Bela Gipp[2,3]
[1]DFKI GmbH, Germany (firstname.lastname@dfki.de)
[2]University of Konstanz, Germany (firstname.lastname@uni-konstanz.de)
[3]University of Wuppertal, Germany (lastname@uni-wuppertal.de)

ABSTRACT

Many digital libraries recommend literature to their users considering the similarity between a query document and their repository. However, they often fail to distinguish what is the relationship that makes two documents alike. In this paper, we model the problem of finding the relationship between two documents as a pairwise document classification task. To find the semantic relation between documents, we apply a series of techniques, such as GloVe, Paragraph-Vectors, BERT, and XLNet under different configurations (e.g., sequence length, vector concatenation scheme), including a Siamese architecture for the Transformer-based systems. We perform our experiments on a newly proposed dataset of 32,168 Wikipedia article pairs and Wikidata properties that define the semantic document relations. Our results show vanilla BERT as the best performing system with an F1-score of 0.93, which we manually examine to better understand its applicability to other domains. Our findings suggest that classifying semantic relations between documents is a solvable task and motivates the development of recommender systems based on the evaluated techniques. The discussions in this paper serve as first steps in the exploration of documents through SPARQL-like queries such that one could find documents that are similar in one aspect but dissimilar in another.

CCS CONCEPTS

• **Information systems** → *Recommender systems*; *Similarity measures*; *Clustering and classification*; • **Computing methodologies** → *Supervised learning by classification.*

KEYWORDS

document similarity, recommender systems, document classification, Siamese networks, Transformers, BERT, XLNet, Wikipedia

1 INTRODUCTION

To cope with the ever-emerging information overload, digital libraries employ literature recommender systems (LRS) [7]. These systems recommend related documents with the help of similarity measures, which often only distinguish between similar and dissimilar documents. This simplification neglects the many facets of extensive documents in digital libraries. It remains unclear to which of the many facets the similarity relates. In philosophy [18], but also in natural language processing (NLP) [6], the similarity of A to B has been addressed as an ill-defined notion unless one can say to what the similarity relates. For LRS, one would rather know what aspects of the two documents are similar or how they relate to each other than just knowing that the documents are similar or dissimilar. Identifying the aspects connecting different documents would allow users to explore the document space by formulating SPARQL-like queries in terms of documents and their relations (e.g., find a document with one specific relation to A, but a different relation to B). These queries are generally referred to as analogical queries [17]. Especially for complex information needs, the formulation of analogical queries is more intuitive [24]. A system that supports analogical queries would be particularly beneficial for scientific literature since the discovery of the analogies is crucial for scientific progress [10].

Nonetheless, document similarity measures do not take into account the semantic relations that would underpin such a system. While other NLP tasks, like relation extraction (RE) [42], deal with relations, they are not concerned with semantic relations between documents. For instance, RE is about relations between entities occurring within a single document text. Similarly, the document classification task aims to categorize individual documents, but fail to address the relationship that binds two or more documents.

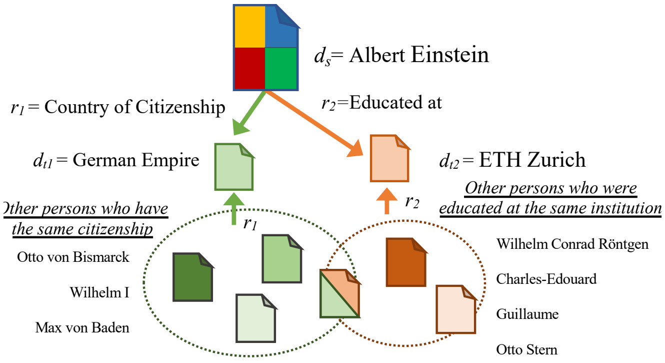


Figure 1: Semantic relations between Wikipedia articles. Seed article *Albert Einstein* is connected to other articles by the property *educated at* and *citizenship*. Considering articles only a single edge apart leads to diverse recommendations, while two edges can be utilized for recommendation sets focused on a specific or an intersection of aspects.

In this paper, we combine the ideas of relation extraction, document classification, and document similarity to classify the semantic relation of document pairs. Given a seed document d_s, we are interested in finding a target document d_t that shares the semantic relation r_i with d_s. We use the term "semantic relation" to indicate connections between two documents above the syntax level [21]. We model the task of finding the relation r of a document pair (d_s, d_t) as a pairwise multi-class document classification problem.

The semantic relation between documents provides context for similarity and enables analogical queries. To evaluate the presented techniques, we build a dataset using Wikipedia and Wikidata [38] repositories to illustrate our problem. Wikipedia articles are the seed and target documents, while Wikidata properties provide the semantic relations between a document pair. Figure 1 shows one example from our dataset. The articles *Albert Einstein* and *German Empire* are the pair (d_s, d_{t_1}) and the relation is defined by r_1, which is the Wikidata property *country of citizenship*. These relations enable recommendations and analogical queries (Section 5).

Our paper makes three major contributions. First, we propose a method to classify the semantic relations of document pairs. Second, we implement six different models using word-based document embeddings from GloVe [31] and Paragraph Vectors [23] (as Doc2vec implementation [33]), and deep contextual language models from BERT [15] and XLNet [41] in a vanilla and Siamese architecture [9]. Each system is evaluated under specific configurations regarding its concatenation method and sequence length. Third, we introduce a novel dataset composed of 32,168 Wikipedia article pairs and Wikidata properties that define the semantic relation of these articles. All our datasets, trained models, and source code are publicly available to contribute to transparency and reproducibility.

2 RELATED WORK

In the following, we relate to other research regarding document similarity and its use for recommender systems and analogies. Besides, we refer to related work that applies similar or the same techniques for solving other NLP tasks.

2.1 Document Similarity & Recommendations

Bär et al. [6] discuss the notion of similarity between texts in the context of NLP. They express that while text similarity is present in many NLP tasks, the similarity is often ill-defined and used as an "umbrella term covering quite different phenomena". Bär et al. [6] formalize text similarity and suggest content, structure, and style as the major dimensions inherent to texts. With approximately 55% of publications using content-based filtering, it accounts for the majority of the LRS research [7]. Structure and style are not actively being accounted for. Therefore, we focus only on the content.

Giving its diversity and reach, Wikipedia had been used as a laboratory in which recommender system methodologies can be tested [28, 36]. In [36], we compared text- and link-based document similarity measures and found that both methods capture similarity differently. Link-based methods tend to retrieve documents from a broader context, while text-based methods are focused on specific terms and topics. Consequently, each similarity approach is suitable for different information needs, e.g., getting an overview of a topic or performing in-depth research. With the classification of semantic document relations, we intend to tailor recommendations depending on specific information needs. For example, we could provide either recommendations focusing on a particular relation class or diverse recommendations from multiple relation classes.

2.2 Analogical Queries

An analogy is a comparison between two or more elements in which their relation is used to illustrate an explanation. Moreover, analogical query solving in the form of "A is to B as C is to ?" is a fundamental aspect of human intelligence [17, 24]. Chan et al. [10] emphasize the importance of analogical query solving for scientific progress. They propose a semi-automated approach for finding analogies between research papers using expert and crowd annotators to segment the abstracts of papers into background, purpose, mechanism, and findings. Next, they encode the segments with GloVe [31] and Paragraph Vectors [23] and compute their similarity to determine whether papers are similar with respect to those segments. However, segmentation breaks the coherence of documents. Our method aims to find semantic relations between documents while maintaining their coherence intact.

In the context of word embeddings, analogies are often illustrated using vector arithmetic, e.g., $\vec{w}_{\text{King}} - \vec{w}_{\text{Queen}} = \vec{w}_{\text{Man}} - \vec{w}_{\text{Woman}}$ [25]. Allen and Hospedales [3] give a mathematical description of analogies as linear relationships between word embeddings. Dai et al. [13] demonstrate that such analogies are also present in document embeddings. In their experiment, using Wikipedia articles, the nearest neighbor to the vector of $\vec{w}_{\text{LadyGaga}} - \vec{w}_{\text{American}} + \vec{w}_{\text{Japanese}}$ is the article on Ayumi Hamasaki, a famous Japanese singer that published an album called "Poker Face" in 1998 (like Lady Gaga in 2008).

2.3 Transformers

Recently, Transformer-based [37] neural language models introduced a shift from context-free word embeddings, like GloVe [31], to contextual embeddings as the ones used in BERT [15] and XLNet [41]. The Transformer architecture allowed the efficient unsupervised pretraining of language models and led to significant improvements in many NLP benchmarks [26, 39, 43]. Reimers and Gurevych [34] proposed to combine BERT with a Siamese architecture [9] for semantic representations of sentences and their similarity [26]. In prior work [32], we also utilized a Siamese BERT model to determine the discourse relations between text segments to generate a story for the segments. Moreover, BERT has successfully solved various document classification tasks [1, 29]. Akkalyoncu Yilmaz et al. [2] apply BERT to an information retrieval system for an end-to-end search over large document collections. Despite their success in NLP, Transformers have gained little attention in the recommender system community so far and are not even mentioned in a recently published survey [5]. To our knowledge, Hassan et al. [27] are one of the first to use BERT to recommend research papers. As opposed to our work, Hassan et al. use BERT to encode only the paper titles as vectors and then generate recommendations using cosine similarity. In our experiments, we utilize the article text and learn the document relation using a multilayer perceptron (MLP).

3 METHODOLOGY

In the following, we describe the dataset and investigated systems to facilitate the reproduction of our results.

3.1 Data set & Use case

Existing datasets provide either classifications of single documents (e.g., topic [29]), relations between sentences or entities (e.g., natural language inference [39], word analogies [25], entity relation extraction [42]), or similarity between text pairs (i.e., binary classification [16]). Our task is defined as as multi-class classification of document pairs consisting of multiple sentences. Moreover, the learning characteristic in our task requires considerably larger dataset than [10] or [20]. To the best of our knowledge, no established dataset fulfills these requirements.

3.1.1 Training data. One example of a digital library that employs an LRS is Wikipedia. Recommendations for Wikipedia articles have been addressed in the literature [28, 36]. Wikipedia is connected with Wikidata, an open knowledge graph in which nodes represent items (e.g., Wikipedia articles) and edges represent properties of these items (e.g., relation that connect two different articles). The link of most Wikipedia articles to their corresponding Wikidata items allows the construction of a large dataset tailored to the problem of semantic relation classification. The triple (d_s, d_t, r_i) of two documents d_s and d_t, and the relation class r_i describes a document pair relation. In the Resource Description Framework (RDF) terminology, d_s is the subject, d_t the object, and r_i the predicate, whereas in the Wikidata terminology, a relation corresponds to a statement[1]. The relation class r_i (predicate) is a Wikidata property that semantically relates a pair of Wikipedia articles (d_s, d_t). For instance, the Wikipedia article of *Albert Einstein*[2] and its Wikidata item[3] is connected to the article[4] and item[5] of the *German Empire* through the property *country of citizenship*[6]. The Wikidata property acts as both, the relation of the Wikipedia article pair and the class label in the training data for this same pair of documents. Table 1 lists other examples to illustrate our scenario better.

Given Wikipedia's nature as an encyclopedia, its use as the dataset has some shortcomings. Encyclopedic documents tend to describe a single entity, and their semantics can be seen as rather homogeneous in comparison to other literature forms. Nonetheless, we consider Wikipedia and Wikidata to be a suitable corpus to demonstrate our approach. Wikidata properties range from entity-specific relations (e.g., *educated at*) to abstract ones (e.g., *facet of*). Wikipedia articles and their relations are, on average, more comprehensible than those in scientific literature , which contributes to the analysis of our results. Another aspect that supports our choice of Wikipedia and Wikidata is their open license copyright.

3.2 Semantic Relations

At the time of writing, Wikidata contained 7,091 properties[7] of which we selected the following nine for this research:

- *country of citizenship* - seed is citizen of the target;
- *different from* - item that is different from another item, with which it is often confused;
- *educated at* - educational institution attended by seed;
- *employer* - seed works or worked for target;
- *facet of* - topic of which this item is an aspect, item that offers a broader perspective on the same topic;
- *has effect* - the seed causes the target;
- *has quality* - the entity has an inherent or distinguishing non-material characteristic;
- *opposite of* - item that is the opposite of this item;
- *symptoms* - possible symptoms of a medical condition.

Table 1 lists the corresponding Wikidata PIDs, their quantity, and examples for each property. Besides the number of available Wikipedia article pairs, diversity was also a criterion in our selection. Diversity refers to the different semantic meanings of properties (e.g., *country of citizenship*, *opposite of*). Similarly, the requirements to predict a relation between documents can also be diverse. While some relations are clearly expressed within the document text (e.g., for documents referencing people, their citizenship is often put in the first sentences), others will require a more comprehensive understanding of the article content. For instance, while *floor* as the opposite of *ceiling* is evident, this fact will most likely not be explicitly mentioned in the article text. Also, other relations like *has effect* or *symptoms* can require unwritten domain knowledge. The classification performance can also be affected by the type of the connected articles. For example, the relation class *country of citizenship* exclusively connects persons and countries. No other property uses such a combination. On the contrary, the relation classes *educated at* and *employer*, connect a person with an organization. Additionally, all relations are unidirectional, except for *opposite of.* Given the many aspects our relations are exploring, we expect significant differences in the classification performance.

Table 1: The relation classes with their Wikidata PIDs, three examples, and the number of samples in our dataset.

Relation class	PID	#	Example relations
country of citizenship	P27	3636	Torben Ulrich ⟶ Denmak Neal Doughty ⟶ United States Julian Kenny ⟶ Trinidad and Tobago
different from	P1889	4048	Computer file ⟶ File folder Lee County, Alabama ⟶ Lee County, Illinois Karo ⟶ Karo (name)
educated at	P69	1798	Hillar Eller ⟶ University of Tartu Al Young ⟶ University of Michigan Heinrich Finkelstein ⟶ Leipzig University
employer	P108	1557	Gary M. Mavko ⟶ Stanford University Alexander Medvedev ⟶ Gazprom John Reif ⟶ Duke University
facet of	P1269	1343	Reformation ⟶ Protestantism 1974 in Portugal ⟶ Portugal Sportsmanship ⟶ Sport
has effect	P1542	698	Language attrition ⟶ Extinct language Arsenic poisoning ⟶ Lung cancer Foul ball ⟶ Out (baseball)
has quality	P1552	1022	Antisemitism ⟶ Nazism Employment ⟶ Access badge Human ⟶ Gender
opposite of	P461	929	Floor ⟶ Ceiling Person ⟶ Society Exponentiation ⟶ Logarithm
symptoms	P780	1053	Myalgia ⟶ Influenza Mercury poisoning ⟶ Cough Death rattle ⟶ Sound
	Total	16,084	

[1] https://www.wikidata.org/wiki/Help:Statements
[2] https://en.wikipedia.org/wiki/Albert_Einstein
[3] https://www.wikidata.org/wiki/Q937
[4] https://en.wikipedia.org/wiki/German_Empire
[5] https://www.wikidata.org/wiki/Q43287
[6] https://www.wikidata.org/wiki/Property:P27
[7] https://tools.wmflabs.org/hay/propbrowse/

3.3 Data Preprocessing

We sampled 10,000 article pairs in total with a balanced class distribution over the nine properties. The relations were obtained trough the Wikidata SPARQL interface in December 2019. For each Wikipedia article in the sample, we also checked whether the article was connected to any other article but was not part of the initial sample and retrieved the missing relations. We removed all duplicated article pairs and multi-label relations. The main goal of this paper is to explore the multi-class classification problem, so we ensure that the same pair of documents did not share different labels. Wikidata provides data for multi-label relations, especially for hierarchical properties. However, only less than 1% of our sample data contained multi-label relations. For the sake of simplicity, we decided to remove them. This procedure generates 16,084 Wikipedia article pairs with an imbalanced class distribution (Table 1). The increase of samples is due to the retrieval of missing relations. The corresponding articles were converted to plain-text from the English Wikipedia dump of November 2019 using the Gensim API [33].

3.4 Negative Sampling

In addition to the nine positive classes from Wikidata, we introduce a class named *None* that works as negative cases for our positive samples in the same proportion. The articles in the *None* category are randomly selected and do not share any relation with the positive ones. The resulting dataset contains 32,168 samples in total.

3.5 Systems

This paper evaluates six classifiers under different configurations, totaling 30 systems. We distinguish between three model categories: (i) document embeddings from word embeddings using the full document text (GloVe and Doc2vec), (ii) Vanilla Transformers, and (iii) Siamese Transformers (each Transformer as BERT and XLNet). Each classifier takes two documents d_s and d_t as input and predicts their relation $\hat{y} = \text{rel}(d_s, d_t)$ as its output. The hyperparameters for the considered systems are detailed in Section 3.6.

3.5.1 Doc2vec. With word2vec, Mikolov et al. [25] introduced an algorithm to learn dense vector representations of words such that semantically similar words end up close to each other in the embedding space. Word2vec is widely applied in NLP tasks [19, 35] but unable to represent entire documents. Paragraph Vectors [23] (also known as Doc2vec), extends word2vec to learn embeddings for word sequences of arbitrary length. In the following, we refer Paragraph Vectors as Doc2vec, since we employ the widely-used implementation of the Gensim [33] framework. We obtained a 200D document vectors $\vec{d}$ for each Wikipedia article by training Doc2vec's distributed bag of words model (dbow) using both training and test data, and the default hyperparameters in Gensim[8]. The document vector size of 200 corresponds to the size of the GloVe word vectors (Section 3.5.2). The choice of dbow over the distributed memory training model is due to its results in semantic similarity tasks [22]. It is important to mention that even though the embeddings model used both training and test sets, the latter was not used for training the classifier.

3.5.2 AvgGloVe. GloVe [31] also produce dense embedding representations, but unlike word2vec, GloVe is a count-based method that uses global statistics to derive its word vectors. In GloVe, the co-occurrence matrix explores the ratio of the probabilities of words in a text to derive its semantic vectors. While we use the 200D pretrained word embedding model[9], GloVe does not provide document vectors directly. To embed a Wikipedia article $\vec{d}$, we compute the weighted average over its word vectors $\vec{w}_i$ (AvgGloVe), whereby the number of occurrences of the word i in d defines the weight c_i. Arora et al. [4] showed the weighted average of word vectors is effective and yields good results for representing documents.

For both full-text methods, AvgGloVe and Doc2vec, we encode each document from our document pair (d_s, d_t) independent from the classification task and concatenate their resulting vectors. The different concatenation variants tested in our experiments are discussed in Section 3.6. The resulting document pair vector is then used as an input to a fully-connected MLP, which classifies the document pair relation $\hat{y} = \text{rel}(d_s, d_t)$. The dimension of the output of the last layer of all classifiers $(\hat{y})$, corresponds to the nine Wikidata properties (Table 1) and one additional dimension for the *None* class of negative samples (Section 3.4). The logistic sigmoid function is used to generate the probabilities for the multi-class classification.

3.5.3 Vanilla Transformer. As the third model category, we employ two language models for deep contextual text representations based on the Transformer architecture [37], named BERT [15] and XLNet [41]. The two Transformer models are originally designed to solve sequence pair classification. The base training task (i.e., next sentence prediction) for BERT and XLNet allows us to fine-tune them for the document pair classification task. The content of the document pair (i.e., title and text of d_s and d_t) is tokenized, delimited with special tokens, i.e., `[CLS]` and `[SEP]` for BERT, `<cls>` and `<sep>` for XLNet, and then jointly fed trough the Transformer (Figure 2). The Transformer output is used as the input to a single fully-connected linear layer with 512 units for the classification (prediction head). Regarding terminology, we refer to the two models as vanilla Transformer since their original architecture is unchanged.

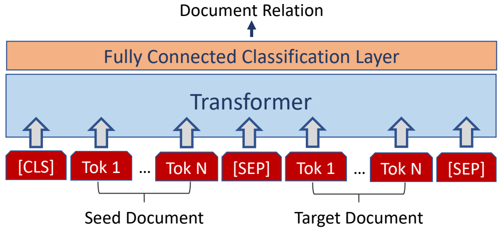


Figure 2: Vanilla Transformer for sequence pair classification. `[SEP]`-token separates seed and target document.

3.5.4 Siamese Transformer. We combine the two Transformers (BERT and XLNet) in a Siamese network architecture [9]. In Siamese networks, two inputs are fed through identical sub-networks with shared weights (in this case, the Transformers), and then passed to a classifier or a similarity function. Reimers and Gurevych [34] have

[8] https://radimrehurek.com/gensim/models/doc2vec.html

[9] https://nlp.stanford.edu/projects/glove/

shown that Siamese BERT networks are suitable for text similarity tasks. For our experiment, both documents d_s and d_t are input to the Transformer sub-networks to derive two contextual document vectors (Figure 3). Next, the document vectors are concatenated and classified with a 2-layer MLP (2x512 units with ReLU activation), the same method applied by Doc2vec and AvgGlove. In contrast to Doc2vec and AvgGloVe, the document representations are neither fixed nor frozen, but continually learned during the training of the classifier. Different than [34], our implemented Siamese architecture is applied to a multi-class classification instead of a binary one.

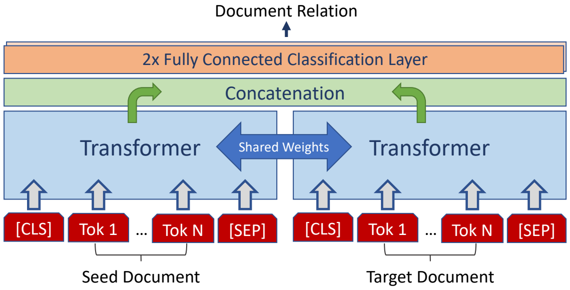


Figure 3: Siamese Transformer architecture. Both documents fed separately trough the Transformer, the concatenated document vectors are input to the classification layer.

The architectures of the underlying BERT and XLNet models are the corresponding BASE-CASED versions of the pretrained models with 12-layer, 768-hidden, 12-head, and 110M parameters. Even though the architectures of BERT and XLNet are comparable, the associated language models are pretrained with different data. While BERT is trained on English Wikipedia and the BooksCorpus [43] alone, XLNet uses additional Web corpora for pretraining [41].

3.6 Hyperparameters

3.6.1 Sequence length. The vanilla and Siamese Transformer models based on BERT have a maximum sequence length of 512 tokens due to absolute positional embeddings. However, XLNet integrates the relative positional encoding, as proposed in Transformer-XL [14]. Therefore, XLNet's architecture is, in theory, not bound to a maximum sequence length. However, a custom pretraining is out of scope for this research, and the publicly available pretrained models of XLNet have the same 512 token limit as BERT. It remains unknown how the length of the processed sequence affects the classification task. From [36], we know that the performance of similarity measures peaks at 450 words since the introduction section in Wikipedia articles presumably contains all essential information. Other sections might add only noise and make it harder to encode relevant semantic information from the articles. Thus, we evaluate the Transformers using 128, 256, and 512 tokens (Section 4.2).

3.6.2 Concatenation. Doc2vec, GloVe, and the Siamese models concatenate the separately encoded document vectors $\vec{d_s}$ and $\vec{d_t}$. In the literature, there is no widely accepted concatenation method. For instance, Conneau et al. [12] use $[u; v; |u - v|; u * v]$ for sentence embedding, while Sentence-BERT [34] presents $[u; v; |u - v|]$ as the best method. In Section 4.3, we test the following variations:

- $[u; v]$ Concatenation of the two vectors u and v;
- $[u; v; |u - v|]$ and absolute value of element-wise difference;
- $[u; v; |u - v|; u * v]$ and element-wise product.

3.7 Implementation

All experiments with Doc2vec and AvgGloVe can be run on CPU in less than 15 minutes using the Gensim [33] and Scikit-learn [30] framework. Before training the Doc2Vec model, Gensim preprocesses[10] the plain-text from the Wikipedia articles. For AvgGloVe, the individual words occurring in the article text are extracted with Scikit-learn's CountVectorizer[11] including English stop word removal. The Transformer models require a GPU as hardware. We rely on HuggingFace's PyTorch implementation [40] of BERT and XLNet. The training time for a single epoch on a GeForce GTX 1080 Ti (11 GB) ranged from less than 10 minutes for vanilla BERT-128 (simplest Transformer architecture), to 55 minutes for Siamese XLNet-512 (most complex Transformer architecture). As suggested in [15], the Transformer training is performed with batch size $b = 4$, dropout probability $d = 0.1$, learning rate $\eta = 2^{-4}$ (Adam optimizer) and 4 training epochs. If not otherwise stated, the default settings of the frameworks were used. The evaluation is conducted as stratified k-fold cross-validation with $k = 4$ and 24,126 training, and 8,041 test samples (the class distribution remains identical for each fold). The source code, dataset, and trained models are publicly available on Zenodo[12], GitHub[13] and as a demo on Google Colab[14].

4 RESULTS

Our results are divided in: overall, sequence length, concatenation, relation classes, and manual sample examination. These five subsections move from a high-level perspective to a detailed investigation of the main aspects that most contributed to our findings.

4.1 Overall

The empirical results of the tested systems and hyperparameters are presented in Table 2. Vanilla BERT-512 yields the best micro average F1-Score with 0.933, followed by its 256 length size model, with 0.930. The second-best model is the vanilla XLNet-512 with 0.926 F1 and a statistically significant lower score compared to vanilla BERT-512 (95% confidence interval). The vanilla Transformers generally outperform their Siamese counterparts. Siamese BERT (0.870 F1) and Siamese XLNet (0.870 F1) do not achieve the same performance as their vanilla architectures for the same 128 sequence length size, with scores of 0.920 (BERT-128) and 0.914 (XLNet-128) respectively. The shared contextual information during the encoding of document pairs most likely yields the better performance of vanilla Transformers. AvgGloVe (0.875 F1) outperforms Siamese BERT and Siamese XLNet, which makes AvgGloVe preferable over Siamese Transformers since AvgGloVe requires only a fraction of the computing resources and runs on commodity hardware. With

[10] https://radimrehurek.com/gensim/utils.html#gensim.utils.simple_preprocess
[11] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[12] https://doi.org/10.5281/zenodo.3713183
[13] https://github.com/malteos/semantic-document-relations
[14] https://ostendorff.org/r/jcdl2020-colab

Table 2: Results as micro avg. F1-score with standard deviation in 4-fold cross-validation for all system configurations including full-text document embeddings from GloVe and Doc2vec, and vanilla and Siamese Transformers (BERT-base and XLNet-base). Vanilla BERT-512 performs best.

Model	Seq.	Concatenation	F1	Std.
AvgGloVe	-	$u; v$	0.863	± 0.0040
		$u; v; \|u-v\|$	0.871	± 0.0045
		$u; v; \|u-v\|; u*v$	0.875	± 0.0036
Doc2vec	-	$u; v$	0.838	± 0.0049
		$u; v; \|u-v\|$	0.836	± 0.0048
		$u; v; \|u-v\|; u*v$	0.845	± 0.0019
Siamese BERT	128	$u; v$	0.844	± 0.0025
		$u; v; \|u-v\|$	0.859	± 0.0080
		$u; v; \|u-v\|; u*v$	0.856	± 0.0102
	256	$u; v$	0.851	± 0.0046
		$u; v; \|u-v\|$	0.860	± 0.0137
		$u; v; \|u-v\|; u*v$	0.862	± 0.0090
	512	$u; v$	0.846	± 0.0050
		$u; v; \|u-v\|$	0.860	± 0.0087
		$u; v; \|u-v\|; u*v$	0.870	± 0.0067
Siamese XLNet	128	$u; v$	0.855	± 0.0075
		$u; v; \|u-v\|$	0.869	± 0.0061
		$u; v; \|u-v\|; u*v$	0.867	± 0.0068
	256	$u; v$	0.856	± 0.0106
		$u; v; \|u-v\|$	0.869	± 0.0071
		$u; v; \|u-v\|; u*v$	0.870	± 0.0078
	512	$u; v$	0.856	± 0.0110
		$u; v; \|u-v\|$	0.860	± 0.0179
		$u; v; \|u-v\|; u*v$	0.864	± 0.0096
Vanilla BERT	128	-	0.920	± 0.0028
	256	-	0.930	± 0.0042
	512	-	**0.933**	± 0.0039
Vanilla XLNet	128	-	0.914	± 0.0065
	256	-	0.914	± 0.0023
	512	-	0.926	± 0.0016

an F1-score of 0.845 at its best configuration, Doc2vec is the worst performing model. In summary, we consider the results of AvgGloVe and vanilla BERT as most promising for future application scenarios. We hypothesize that an F1-score of above 0.90 is already suitable enough for LRS. Especially, expert users would tolerate some misclassifications in favor of otherwise undiscoverable information. This would be the case for target documents that are considered to be dissimilar to the seed with existing methods but are found to have semantic relation with the help of our methods.

4.2 Sequence Length

As explained in Section 3.6, we are particularly interested in the effect of the sequence length on the Transformer models. To illustrate this effect, Figure 4 shows the comparison of Siamese BERT, Siamese XLNet, vanilla BERT, and vanilla XLNet with respect to their sequence length (i.e., 128, 256, and 512). In this comparison, the Siamese models use the best performing concatenation method, which is $[u; v; |u-v|; u*v]$. Our findings reveal that longer sequences are related to better results. For all models, except Siamese XLNet, the highest F1-score is achieved with 512 tokens and the second-highest with 256 tokens. One could think this outcome is to be expected. However, in [36], the performance of text- and link-based document similarity measures declines for Wikipedia articles with more than 450 words. When comparing Siamese with vanilla Transformers, the vanilla models work with only half of the sequence length to encode one document of the pair. In vanilla Transformers, the document pairs share the sequence length, while in Siamese Transformers each document has its own Transformer subnetwork (sequence length). For example, a vanilla 128-Transformer would use only 62 or 63 tokens of each document (three tokens are reserved for special tokens as Figure 2 shows). Thus, the small performance difference within vanilla BERT with 512 tokens (0.933 F1), 256 tokens (0.930 F1), and 128 tokens (0.920 F1) is remarkable. Moreover, the performance differences should be considered relative to the higher computation expenses of longer sequences.

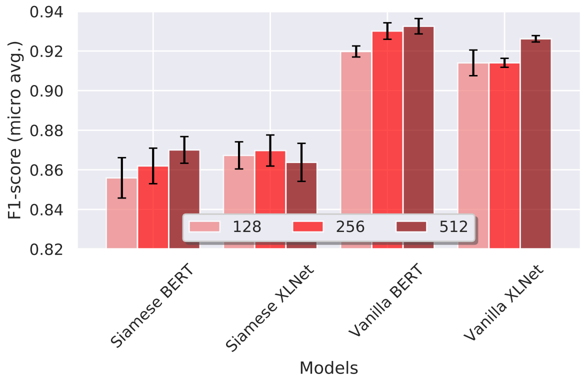


Figure 4: Performance of vanilla and Siamese Transformers w.r.t. sequence length. Siamese models use $[u; v; |u-v|; u*v]$ as concatenation. Aside from Siamese XLNet, 512 tokens achieve the best F1-scores for all models.

4.3 Concatenation

Aside from the sequence length, we also analyzed the different concatenation methods in AvgGloVe, Doc2vec, and the Siamese models (Figure 5). All models achieve the highest F1-score when the concatenation with an element-wise difference and product is used ($[u; v; |u-v|; u*v]$). Furthermore, we confirmed the results of Reimers and Gurevych [34], i.e., the most crucial component is the element-wise difference $|u-v|$. Only for Doc2vec the element-wise difference decreases the performance in comparison to the simple concatenation. However, this performance decrease is marginal and within standard deviation. In general, the element-wise difference measures the distance between the dimensions of the two document vectors and, thus, ensures that similar pairs are closer to each other than dissimilar pairs. This effect is evident for Siamese BERT and Siamese XLNet, for which the element-wise difference yields the most substantial performance improvement. On the contrary, the element-wise product adds only a small improvement to our models.

Table 3: Results for precision (P), recall (R), F1-score, and sample count in test data w.r.t. relation classes. Evaluated systems are AvgGloVe, Siamese BERT, vanilla BERT and vanilla XLNet. The results of other models are published along with the code.

Model	**AvgGloVe**			**Siamese BERT-512**			**Vanilla BERT-512**			**Vanilla XLNet-512**			
Relation class	P	R	F1	P	R	F1	P	R	F1	P	R	F1	Samples
country of citizenship	0.963	0.983	0.973	0.956	**0.996**	0.976	**0.993**	**0.996**	**0.994**	0.989	**0.996**	0.993	909
different from	0.856	0.843	0.849	0.872	0.899	0.885	**0.971**	0.931	**0.950**	0.969	**0.933**	**0.950**	1012
educated at	0.683	0.729	0.703	0.730	0.740	0.734	0.759	**0.900**	**0.817**	**0.774**	0.759	0.763	450
employer	0.662	0.620	0.639	0.639	**0.769**	0.695	**0.892**	0.653	**0.740**	0.711	0.748	0.725	389
facet of	0.786	0.781	0.782	0.839	0.785	0.810	**0.916**	**0.908**	**0.911**	0.888	0.904	0.896	336
has effect	0.644	0.606	0.620	0.626	0.468	0.502	**0.783**	0.614	0.683	0.768	**0.658**	**0.704**	175
has quality	0.694	0.682	0.687	0.662	0.619	0.639	0.718	0.797	0.749	**0.763**	**0.799**	**0.774**	256
opposite of	0.672	0.666	0.667	0.540	0.791	0.640	0.761	0.763	0.756	**0.773**	**0.835**	**0.795**	232
symptoms	**0.887**	0.932	0.908	0.827	0.969	0.892	0.872	0.973	**0.920**	0.864	**0.984**	0.919	263
none	0.943	0.940	0.942	0.955	0.897	0.925	0.978	**0.981**	**0.979**	**0.979**	0.968	0.973	4021
micro avg	0.875	0.875	0.875	0.870	0.870	0.870	**0.933**	**0.933**	**0.933**	0.926	0.926	0.926	8043
macro avg	0.779	0.778	0.777	0.764	0.793	0.770	**0.864**	0.852	**0.850**	0.848	**0.858**	0.849	8043

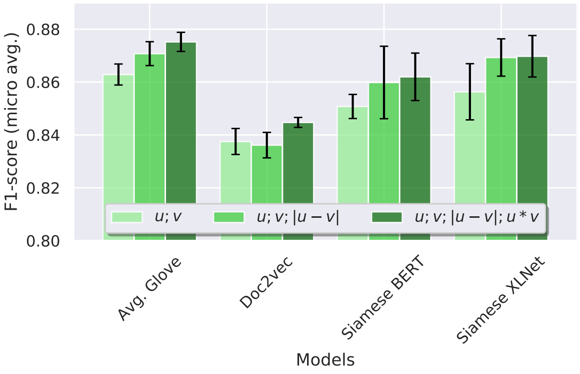
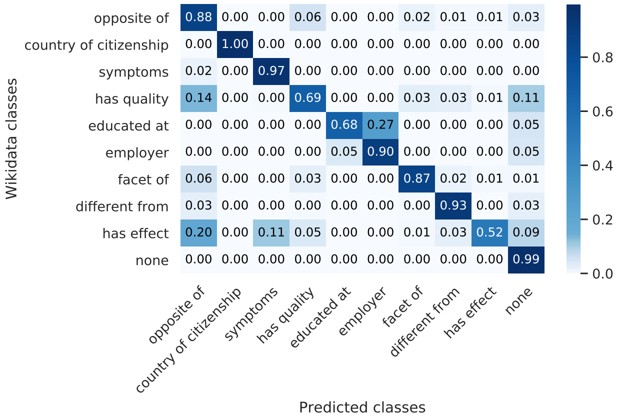


Figure 5: Results of the full-text document embeddings and Siamese Transformer-512 models w.r.t. concatenation.

4.4 Relation Classes

We selected nine diverse Wikidata properties to explore how the systems would respond to the individual challenges of each property. Table 3 presents precision, recall, and F1-score of the best four systems for the different model categories. Each score is the mean over the 4-fold cross-validation (cf. Table 2 for standard deviation). AvgGloVe and Siamese BERT use $[u; v; |u - v; u * v]$ as concatenation method, and all Transformer models (Siamese BERT, vanilla BERT, and vanilla XLNet) use the 512 sequence length. The best relation classes in terms of performance are *country of citizenship*, *none* (negative samples), and *different from*, whereas the classes *employer*, *has quality*, *has effect* yield the lowest scores. Given that the best performing classes are also over-represented in terms of sample count, the outcome suggests that other classes only need more training data. Still, the comparison of the *employer* class (389 test samples, vanilla BERT 0.740 F1) and *facet of* (336 test samples, vanilla BERT 0.911) reveals that the performance difference is also due to the diverse requirements of classes themselves.

The superiority of vanilla BERT is also present in the class-specific evaluation scenario, although it is outperformed by vanilla XLNet for three relation classes with a small number of samples (*has effect*, *has quality* and *opposite of*). In AvgGloVe, *symptoms* has the highest precision score, which is probably caused by AvgGloVe being able to utilize the full-text of articles in contrast to the Transformer models. Medical articles, like *Alcoholism* (Example 9 in Table 4), contain a section "Signs and symptoms" in which their symptoms are listed. However, such a section is not part of the 512 Transformer tokens. When comparing precision and recall for all classes, both scores are mostly balanced. There is only one striking exception for vanilla BERT. For *employer*, the precision score of 0.829 is higher than the recall of 0.653, while for *educated at* the opposite occurs, with a precision of 0.759 and recall of 0.900, but in a smaller magnitude. A reason for this outcome is that *employer* is often confused with *educated at* as Figure 6 shows.



	opposite of	country of citizenship	symptoms	has quality	educated at	employer	facet of	different from	has effect	none
opposite of	0.88	0.00	0.00	0.06	0.00	0.00	0.02	0.01	0.01	0.03
country of citizenship	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
symptoms	0.02	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00
has quality	0.14	0.00	0.00	0.69	0.00	0.00	0.03	0.03	0.01	0.11
educated at	0.00	0.00	0.00	0.00	0.68	0.27	0.00	0.00	0.00	0.05
employer	0.00	0.00	0.00	0.00	0.05	0.90	0.00	0.00	0.00	0.05
facet of	0.06	0.00	0.00	0.03	0.00	0.00	0.87	0.02	0.01	0.01
different from	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.00	0.03
has effect	0.20	0.00	0.11	0.05	0.00	0.00	0.01	0.03	0.52	0.09
none	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99

Figure 6: Confusion matrix for the predicted and Wikidata classes of vanilla BERT. The relation count is normalized. The most frequent confusion is found with the *educated at* and *employer* class for 27% of the test samples.

The confusion matrix in Figure 6 depicts which classes are most often confused with each other. The predicted classes are taken from the vanilla BERT-512 system, whereby the number of true and predicted classifications is normalized to make the different classes comparable. With 27% of the test sample, *educated at* and *employer* are the most mistaken relation classes in our experiments. We see this outcome because both relation classes connect persons and organizations, and we assume it is harder for the classifier to tell the relations apart. For instance, *Albert Einstein* could be employed or educated at *ETH Zurich* (Figure 1). The misclassification between different relations is also found in *opposite of*, *has quality* and *has effect*, which we conclude is because of similar reasons. In particular, the *opposite of* relation connects various types of articles.

4.5 Manual Sample Examination

To validate our empirical findings, we manually examine the prediction from vanilla BERT-512 with a focus on errors (Table 4). Examples 1 and 2 show a desired classifier despite one misclassification according to Wikidata. While *Armenia* is correctly identified as *Rudolf Muradyan's country of citizenship*, *Brazil* is not recognized. However, *Brazil* is also not mentioned in *Rudolf Muradyan's* Wikipedia article. The Wikidata statement is not reflected in the Wikipedia article, which states *Muradyan* as *Armenian* only. Consequently, both predictions would be correct when only considering the article text. Two errors are exemplified in 3 and 4. Even though *Zaki Naguib Mahmoud's* article explicitly expresses the *educated at* relation with the sentence "Mahmoud was educated at Cairo University", *Cairo University* is classified as his *employer.* Despite not being mentioned in *Mahmoud's* article, *King's College London* is also wrongly classified as his *employer.* In example 5, *Light* is incorrectly classified as the quality of *Darkness*, not as opposite of it. Still, *opposite of* is the class with the second-highest probability. Example 6 shows the *Mexican Revolution* as *different from* the *Mexican War of Independence*, which would be clear to a human user since the Wikipedia article contains banner "Not to be confused with the Mexican War of Independence.". However, this banner is missing in the Wikipedia dump and, thus, is not available to the classifier. Many shared terms and vocabulary make their classification hard to predict for the *different from* relation. Examples 7-9 are not discussed, but similarly illustrate the classifier's performance.

Our manual examination confirms the overall results. Most relations are correctly identified, while some relations are missing even if they are explicitly mentioned in the text. An analysis of the inner Transformer components [11] is a subject for future work.

5 DISCUSSION

Given the results in Table 2, we can state that vanilla Transformers outperform all other methods. Rather unexpected is that BERT generally achieves slightly better results than XLNet. According to Yang et al. [41], XLNet surpasses BERT on the related GLUE benchmark [39], so we were expecting a similar outcome. We hypothesize that this difference may be attributed to two reasons, pretraining on different corpora, and smaller models compared to [41]. We use the BASE, not the LARGE versions of the pretrained models used by Yang et al. [41]. Furthermore, the published XLNet BASE model we considered is pretrained on different data than the one in Yang et al. [41][15]. In contrast to BERT, XLNet is pretrained on Web corpora in addition to Wikipedia and the BooksCorpus [43]. The almost exclusive pretraining on Wikipedia most likely causes BERT to surpass XLNet. The effect of domain-specific pretraining on the performance of the language model has already been shown [8].

Our evaluation also shows that the Siamese networks cannot capture the semantic relations as good as vanilla Transformers. In Siamese models, the encoding of the seed document does not affect the target, and vice-versa. Only the MLP is exposed to the documents as a pair in the form of the concatenated document vectors. During the encoding phase, the relation between the documents plays no role. On the contrary, the Multi-Head-Attention mechanism in the vanilla Transformers allows attending on the two documents simultaneously. As the results suggest, this ability is crucial for the pairwise document classification. The Siamese models are also outperformed by the computationally less expensive AvgGloVe. At a general level, the Siamese models are very similar to AvgGloVe (and Doc2vec), since they derive two document vectors and classify their concatenation. So the performance of the method ultimately depends on its ability to encode the documents. Arora et al. [4] have shown that the weighted average of word vectors can outperform more sophisticated methods. AvgGloVe benefits from the fact that it utilizes the full-text article in contrast to the Transformers, which use only the 512 first tokens of the article text. As a result, AvgGloVe is a reasonable method for real production scenarios, in which computational resources are critical concerns. In production scenarios, one would also avoid classifying all possible n^2 document pairs. Instead, evidently unrelated pairs must be filtered out with traditional similarity measures at first.

Regarding the different relation classes, almost all results present reasonable performance. Moreover, complex relation classes like *facet of* or *has effect*, yield promising results, since they are attractive for the recommender system use case. As the example 1 and 2 shows in Table 4, current systems already reveal wrong or contradicting information between Wikidata and Wikipedia. The results suggest that increasing the sequence length beyond the 512 tokens could further improve the Transformer models. Higher sequence length is already possible with XLNet's architecture, but it would require a pretraining step with longer sequences.

From Relations to Recommendations. Classifying the document relations is not a purpose on its own. We envision recommender systems as an example of a downstream task. The obtained relations can be used for diverse or focused recommendations. As the relations describe different facets of the seed document, one could diversify the recommendations. Choosing the recommendations from documents connected with different relation classes to the seed document would ensure diversity. In Figure 1, the *German Empire* and *ETH Zurich* can be considered as diverse recommendations, since they present different aspects of *Albert Einstein*, i.e., his citizenship and education. When considering documents that are connected to a seed (i.e., one common document) over two edges (i.e., different relations), recommendations focusing on specific aspects are more feasible. Diverse and focused recommendations could be especially suitable for scenarios in which different perspectives are required for the same seed. In contrast to user-based recommender systems, content-based approaches usually struggle to account for specific preferences from their users. One way to respect different information requirements would be to suggest

[15] See https://github.com/zihangdai/xlnet#released-models "*This model (XLNet-Base) is trained on full data (different from the one in the paper)*".

Table 4: Example relations between Wikipedia article pairs (seed and target) as defined by Wikidata and as predicted by Vanilla BERT-512 with the first and second highest probability. Correct predictions are marked with ✓.

ID	Seed	Target	Wikidata Relation	1st Prediction	2nd Prediction
1	Rudolf Muradyan	Brazil	country of citizenship	none	country of citizenship ✓
2	Rudolf Muradyan	Armenia	country of citizenship	country of citizenship ✓	none
3	Zaki Naguib Mahmoud	Cairo University	educated at	employer	educated at ✓
4	Zaki Naguib Mahmoud	King's College London	educated at	employer	educated at ✓
5	Light	Darkness	opposite of	has quality	opposite of ✓
6	Mexican Revolution	Mexican War of Independence	different from	has effect	symptoms
7	History of blogging	Blog	facet of	opposite of	different from
8	Iced tea	Ice-T	different from	none	different from ✓
9	Alcoholism	Cirrhosis	has effect	has effect ✓	none

alternative recommendation sets that are focused on specific aspects. In the example of *Albert Einstein*, shown in Figure 1, focused recommendation sets could include articles about people with the same citizenship or the same educational backgrounds. The intersection of relations would even allow finding people with the same citizenship but different educational background. The classification of the document relations, as demonstrated in our experiments, is the foundation for such recommendations.

Generalization. Given the long-term goal of applying the tested methods on non-encyclopedic corpora, the question arises whether our findings are generalizable. We acknowledge that Wikipedia is a presumable a simpler corpus compared to other literature domains like research papers. Wikipedia articles represent distinct entities and most relations are explicitly expressed in the article text. However, even research papers express semantic relations in their abstracts, e.g., "we used X" or "we found Y". Accordingly, we hypothesize that our systems would yield worse but still satisfactory results under comparable conditions (size of training data, pretraining on in-domain corpus etc.). A reference value would be the F1-score of 0.65, which was achieved by SciBERT on the related task of citation intent classification [8]. While the effort for the unsupervised pretraining of a language model is reasonable, we recognize the annotation of sufficient training data for other corpora is one of the most challenging tasks. After all, even annotations can be solved efficiently as Chan et al.'s crowdsourcing approach demonstrates [10]. We are confident that our results are transferable to other domains.

6 CONCLUSION AND FUTURE WORK

This paper introduces the pairwise document classification to determine semantic relations between documents as an underlying task to advance LRS and other information retrieval applications. We elaborate on why document similarity measures do not account for the heterogeneous semantics of extensive documents and argue that similarity needs a context which defines to what it relates.

The task of finding semantic document relations is implemented as a multi-class classification of document pairs. We demonstrate the viability of this approach with a new proposed dataset of 32,168 Wikipedia article pairs and Wikidata properties that define semantic relations among these articles. In an empirical study, we implement six different models AvgGloVe, Doc2vec, Siamese BERT, Siamese XLNet, vanilla BERT and vanilla XLNet, and evaluate them under different settings regarding the concatenation method and sequence length (Table 2). Our evaluation indicates a sequence length of 512 tokens as the best performing sequence limit for the Siamese and vanilla Transformer models. In addition, we identify $[u; v; |u - v|; u * v]$ as the best concatenation method for AvgGloVe, Doc2vec and the Siamese Transformer models. With the manual sample examination and our evaluation for different relation classes, we show the behavior of the classifiers when exposed to different input data and provide an analysis over different perspectives. Moreover, the manual analysis confirms our empirical results.

Our findings suggest that pairwise document classification is a solvable task using existing techniques. Even abstract semantic relations, like *facet of*, yield a considerable high F1-score. This outcome motivates us to investigate the semantic relations between documents of other literature domains, primarily scientific papers. We envision a system that enables users to explore scientific literature in an analogical manner. For instance, users could retrieve other research papers with a similar methodology but different result. Analogies could be even found with programmatic and SPARQL-like queries. To develop such a system, the Open Research Knowledge Graph [20] could be utilized as the scientific equivalence of Wikidata, while research paper from any open digital library, e.g., arXiv, would correspond to Wikipedia articles. Lastly, the presented Wikipedia and Wikidata dataset also facilitate the evaluation of methods in terms of required training data. The estimation the necessary amount of data would a prerequisite for looking into other domains.

ACKNOWLEDGMENTS

The research presented in this article is funded by the German Federal Ministry of Education and Research (BMBF) through the project QURATOR (Unternehmen Region, Wachstumskern, no. 03WKDA1A).

REFERENCES

[1] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, Jimmy Lin, and David R Cheriton. 2019. DocBERT: BERT for Document Classification. (2019). arXiv:1904.08398v1 https://arxiv.org/pdf/1904.08398

[2] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to Document Retrieval with Birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations.* 19–24. https://doi.org/10.18653/v1/D19-3004

[3] Carl Allen and Timothy Hospedales. 2019. Analogies Explained: Towards Understanding Word Embeddings. In *Proceedings of the 36th International Conference on Machine Learning.* 223–231.

[4] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but though Baseline for Sentence Embeddings. In *5th International Conference on Learning Representations (ICLR 2017)*, Vol. 15. 416–424.

[5] Xiaomei Bai, Mengyang Wang, Ivan Lee, Zhuo Yang, Xiangjie Kong, and Feng Xia. 2019. Scientific Paper Recommendation: A Survey. *IEEE Access* 7 (2019), 9324–9339.

[6] Daniel Bär, Torsten Zesch, and Iryna Gurevych. 2011. A reflective view on text similarity. *International Conference Recent Advances in Natural Language Processing, RANLP* September (2011), 515–520.

[7] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries* 17, 4 (2016), 305–338. https://doi.org/10.1007/s00799-015-0156-0

[8] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3613–3618. https://doi.org/10.18653/v1/D19-1371

[9] Jane Bromley, J.W. Bentz, Leon Bottou, I. Guyon, Yann Lecun, C. Moore, Eduard Sackinger, and R. Shah. 1993. Signature verification using a Siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence* 7, 4 (1993).

[10] Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur. 2018. SOLVENT: A Mixed Initiative System for Finding Analogies between Research Papers. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (nov 2018), 1–21. https://doi.org/10.1145/3274300

[11] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP*. 276–286. https://doi.org/10.18653/v1/W19-4828

[12] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 670–680. https://doi.org/10.18653/v1/D17-1070

[13] Andrew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document Embedding with Paragraph Vectors. (2015), 1–8. arXiv:1507.07998 http://arxiv.org/abs/1507.07998

[14] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2978–2988. https://doi.org/10.18653/v1/P19-1285

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186. https://doi.org/10.18653/v1/N19-1423

[16] William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. 9–16.

[17] Mary L. Gick and Keith J. Holyoak. 1983. Schema induction and analogical transfer. *Cognitive Psychology* 15, 1 (1983), 1–38. https://doi.org/10.1016/0010-0285(83)90002-6

[18] N Goodman. 1972. Seven strictures on similarity. *Problems and projects* (1972).

[19] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 897–907. https://doi.org/10.18653/v1/P16-1085

[20] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. (2019), 243–246. https://doi.org/10.1145/3360901.3364435

[21] Christopher S. G. Khoo and Jin-Cheon Na. 2007. Semantic relations in information science. *Annual Review of Information Science and Technology* 40, 1 (sep 2007), 157–228. https://doi.org/10.1002/aris.1440400112

[22] Jey Han Lau and Timothy Baldwin. 2016. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proceedings Workshop on Representation Learning for NLP*. https://doi.org/10.18653/v1/w16-1609

[23] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on Machine Learning* 32 (2014), 1188–1196.

[24] Christoph Lofi and Nava Tintarev. 2017. Towards analogy-based recommendation: Benchmarking of perceived analogy semantics. *CEUR Workshop Proceedings* 1892 (2017), 9–13.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. (2013), 1–12. arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[26] Amita Misra, Brian Ecker, and Marilyn Walker. 2016. Measuring the Similarity of Sentential Arguments in Dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 276–287. https://doi.org/10.18653/v1/W16-3636

[27] Hebatallah A. Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, Alessandro Micarelli, and Joeran Beel. 2019. BERT, ELMo, use and infersent sentence encoders: The Panacea for research-paper recommendation?. In *CEUR Workshop Proceedings*, Vol. 2431. 6–10.

[28] Yann Ollivier and Pierre Senellart. 2007. Finding Related Pages Using Green Measures : An Illustration with Wikipedia. In *Association for the Advancement of Artificial Intelligence Conference*. 1427–1433.

[29] Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, and Georg Rehm. 2019. Enriching BERT with Knowledge Graph Embedding for Document Classification. In *Proceedings of the GermEval 2019 Workshop*.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. https://doi.org/10.3115/v1/D14-1162

[32] Georg Rehm, Karolina Zaczynska, Julian Moreno Schneider, Malte Ostendorff, Peter Bourgonje, Maria Berger, Jens Rauenbusch, Andre Schmidt, and Mikka Wild. 2020. Towards Discourse Parsing-inspired Semantic Storytelling. In *Proceedings of QURATOR 2020 – The conference for intelligent content solutions*.

[33] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. 45–50.

[34] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *The 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*.

[35] Terry Ruas, William Grosky, and Akiko Aizawa. 2019. Multi-sense embeddings through a word sense disambiguation process. *Expert Systems with Applications* 136 (2019), 288 – 303. https://doi.org/10.1016/j.eswa.2019.06.026

[36] Malte Schwarzer, Moritz Schubotz, Norman Meuschke, and Corinna Breitinger. 2016. Evaluating Link-based Recommendations for Wikipedia. *Proceedings of the 16th ACM/IEEE Joint Conference on Digital Libraries (JCDL'16)* (2016), 191–200. https://doi.org/10.1145/2910896.2910908

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems 30* NIPS (jun 2017), 5998–6008.

[38] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57 (2014), 78–85.

[39] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *7th International Conference on Learning Representations, ICLR 2019* (2019), 353–355. https://doi.org/10.18653/v1/w18-5446

[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. (oct 2019). arXiv:1910.03771 http://arxiv.org/abs/1910.03771

[41] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32*. 5754–5764.

[42] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 764–777. https://doi.org/10.18653/v1/P19-1074

[43] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), 19–27. https://doi.org/10.1109/ICCV.2015.11