Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)

More Like This: Semantic Retrieval with Linguistic Information

Steffen Remus[⋆] **Gregor Wiedemann**[⋆†]

Saba Anwar **Fynn Petersen-Frey** **Seid Muhie Yimam** **Chris Biemann**

Universität Hamburg
first.last@uni-hamburg.de

[†]Leibniz-Institute for Media Research | Hans-Bredow-Institute
g.wiedemann@leibniz-hbi.de

Abstract

We investigate the semantic retrieval potential of pre-trained contextualized word embeddings (CWEs) such as BERT, in combination with explicit linguistic information, for various NLP tasks in an information retrieval setup. In this paper, we compare different strategies to aggregate contextualized word embeddings along lexical, syntactic, or grammatical dimensions to perform semantic retrieval for various natural language tasks. We apply this for fine-grained named entities, word senses, short texts, verb frames, and semantic relations, and show that incorporating certain linguistic knowledge improves the retrieval performance over various baselines. In a simulation study, we demonstrate the practical applicability of our findings to speed up the linguistic annotation of datasets. We also show that nearest neighbor classification, which implicitly uses the retrieval setup, works well with only small amounts of training data.[1]

1 Introduction

Neural language models (NLMs) producing contextualized word embeddings (CWEs) such as ELMo (Embeddings from Language Models; Peters et al., 2018), FLAIR (Akbik et al., 2018), or BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2019), or one of its many successors have been a leap forward for multiple NLP tasks. One major reason for this is the fact that current NLMs can generate compositional vector space representations of a word based on the sequential context in which it appears, thus sufficiently representing its compositional meaning. CWEs allow the disambiguation of a word's meaning up to a certain degree, such that, for example, sequence tagging models can distinguish two identical surface forms when used in different contexts. For example, both instances of each of the two words 'can' and 'open' in the following two sentences *"Alice opens the can"* and *"Alice can open the box"* will be represented with quite distinct embeddings. Whereas vectors are expected to be very similar for the word 'open', both representations for 'can' are expected to be inherently different, indicating a syntactic and semantic shift.

Still, although certain dependency relations are implicitly encoded in BERT, no equivalent to holistic parsing of syntactical or grammatical structures can be assumed from BERT's attention mechanism (Htut et al., 2019). We thus hypothesize that downstream NLP tasks benefit from exploiting explicit syntactical and grammatical cues derived from linguistic knowledge in addition to the contextual embeddings. To investigate this hypothesis, we define a set of aggregation strategies for word embeddings along linguistically informed dimensions. Such representations are used to address several downstream tasks: *a*) labeling on the sentence level, where we experiment with *sentiment detection*, *relation identification*, and *semantic frame induction*, and *b*) word-level- and sequence labeling, where we experiment with *named entity recognition* and *word sense disambiguation*.

The explicit use of syntactic information to aggregate CWEs can be regarded as feature extraction or feature transformation. Such features may not only be useful in classification scenarios but also for retrieval tasks. Particularly, they can be useful in the context of a retrieval scenario in which the ultimate goal is to enable users to rapidly find semantically similar word patterns or sentences in their datasets.

In this regard, there are three main contributions of this paper: *a*) We introduce several different strategies to incorporate explicit linguistic informa-

[⋆]Equal contribution

[1]Our code, experiments and results are published as open source software under a permissive Apache v2 license: https://github.com/uhh-lt/cwe-ling

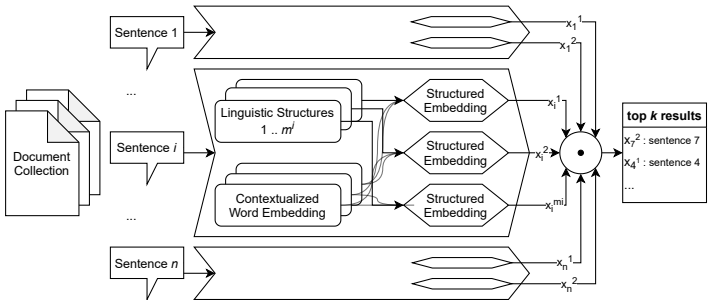


Figure 1: Overview of the retrieval process.

tion for embedding-based feature representations. *b*) We evaluate these strategies in an information retrieval setup to find semantically related items for various downstream NLP tasks. *c*) We demonstrate two potential applications of our findings 1) for speeding up manual annotation of text data, and 2) for fast nearest neighbor classification with little training data. Depending on the task, our retrieval evaluation shows the retrieval precision and nearest neighbor classification indeed profit from the incorporation of additional explicit linguistic knowledge. Depending on the complexity of the task, and correlating it with a simulated cognitive shift between dissimilar texts and distinct categories, our simulation shows that the use of linguistic structures in a retrieval scenario can speed up the manual annotation of text data, e.g. to create training data more rapidly.

2 Related Work

The LISA (linguistically-informed self-attention) approach by Strubell et al. (2018) showed the benefit of injecting syntactic information into a neural network using self-attention for multi-task learning. LISA was applied for dependency parsing, part-of-speech tagging, predicate detection, and semantic role labeling, where the results for all tasks showed significant improvements over the previous state-of-the-art, particularly when using ELMo embeddings (Peters et al., 2018).

Wiedemann et al. (2019) showed that contextual embeddings, particularly BERT (Devlin et al., 2019) inherit a certain degree of sense representation, i.e. polysemous words appear in different areas of the embedding space depending on their context. Wang et al. (2019) implement Elman (1990)'s theory, which states that neural language models are sensitive to word order regularities in simple sentences, by specifically exploiting the inner-sentence structure (word-level ordering) and inter-sentence structure (sentence-level ordering) as training objectives. They argue that their StructBERT model successfully captures the structure of sentences during pre-training.

Htut et al. (2019) and Clark et al. (2019) analyze to which extent attention heads in BERT can track linguistic dependencies. Both works conclude that some attention heads specialize in syntactic structure. Wu et al. (2020) measure the impact one word has on another in a sentence by using a so-called perturbed masking technique. They can derive a syntax tree from a word-word matrix. Soares et al. (2019) used a so-called masking technique to specifically force the model to learn entity locations in a sentence. By doing so, specific representations for particular relations within text can be learned.

SBERT (SentenceBERT; Reimers and Gurevych, 2019) is an extension to pre-trained transformer architectures such as BERT or RoBERTa, which is specifically targeted for sentence similarity search, i.e. finding similar sentences by using cosine similarity. SBERT outperforms most other embedding strategies for multiple sentence similarity tasks. However, it requires labeled data in form of similar/dissimilar sentences.

3 Retrieval of Linguistic Patterns

We approach the problem of semantic retrieval with linguistic structures as follows: Let $\mathrm{S} := [s_1, \ldots, s_n]$ be a dataset with n instances, where s_i represents a sentence. For our retrieval experiments, we use datasets with corresponding class labels $\mathbf{y} = [y_1, \ldots, y_n]$, where y_i is a list of labels in case of word-level tasks. Instances are decomposed into a set of finer-grained, lexical structures such as tokens, multi-word units, chunks, dependency relations, etc. (see Section 3.1 for reference), which we use as the basic unit of retrieval. For each instance s_i, a unique set of m^i linguistic

structures $s_i \mapsto \{\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{m^i}\}$, with replicated y_i labels $\{y_i^1, \ldots, y_i^{m^i}\}$, is extracted by using a particular linguistic pattern. Further, $\mathbf{x}_i^j$ represents a single feature vector extracted by a particular lexical template, for example, it could be the actual sentence embedding or word embedding of s_i. We call $\mathbf{x}_i^j$ a *structured embedding*.

The goal is to retrieve the k most relevant instances for a given query instance q and its extracted *structured embeddings* $q \mapsto \{\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^{m^q}\}$ of a target class c:

$$[r_1, \ldots, r_k] := \underset{i \in \{1 \ldots n\}}{\text{top}_k} \left\{ \underset{\substack{h \in \{1 \ldots m^q\} \\ j \in \{1 \ldots m^i\}}}{\arg\max} \ \text{sim}(\hat{\mathbf{x}}^h, \mathbf{x}_i^j) \right\},$$

where top_k is defined as a function that selects the top k indices as an ordered list from the entire set of labeled instances regarding their maximum similarity score. The sim function is defined to be a similarity function for two vectors; we use *cosine similarity* in our experiments. Figure 1 illustrates the indexing and retrieval process.

3.1 Lexical Structures

For the linguistic pre-processing, i.e. tokenization, part-of-speech tagging (PoS), and dependency parsing we use *spaCy*[2] (unless stated otherwise) and for chunking we use *FLAIR*[3]. For CWEs based on RoBERTa (Liu et al., 2019), we sum the output of the last four layers of the model, and if a token comprises several word piece tokens, the corresponding embeddings are averaged to obtain a single vector for a lexical token. We describe our linguistically informed structures in the following.

3.2 Word-level structures

We use the following two word-level structures to find similar entity spans:

`token`: Each token of the dataset is considered a single item. Consequently, the unit of retrieval is always a single token.

`SPS` (same-PoS-span)**:** In order to capture nouns and noun phrases, each sequence of tokens having the same PoS tag within a sentence is considered as one structure. Thus, the unit of retrieval is a variable-length span of one or more tokens.

3.3 Sentence-level structures

We use the following sentence-level structures to find similar sentences.

[2] https://spacy.io/
[3] https://github.com/flairNLP/flair

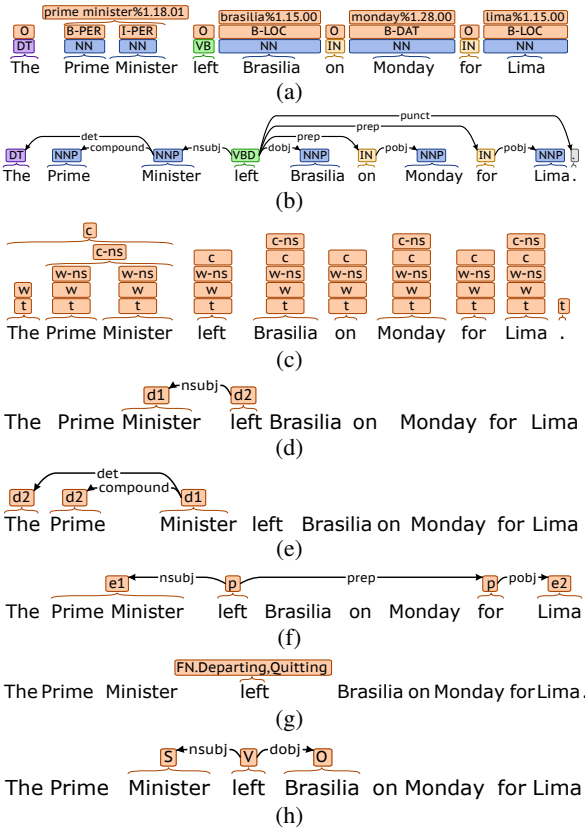


Figure 2: (a) Word-level structures with BIO-labels for NER and WordNet sense information. (b) shows the automatically extracted dependency graph and syntax features. (c-h) Sentence-level structures: (c) shows the aggregation strategy for `token` (t), `word` (w), `word-NS` (w-ns), `chunk` (c), and `chunk-NS` (c-ns). (d) shows the aggregation strategy for `dep-{concat,avg}` for a single dependency edge, i.e. $d1$ and its governor (dependency head) $d2$. (e) illustrates the `dep-depavg` strategy for the word 'Minister', where $d1$ is the actual word and all $d2$ are dependents of $d1$. (f) shows the task dependent `dependency-path` structure for relation identification. (g) and (h) show the task dependent `lexical-unit` and `subj-v-obj` structures for frame identification.

`token`: each token of a sentence is considered a structure.

`word`: same as `token`, w/o punctuation.

`word-NS`: same as `word`, w/o stop-words.

`chunk`: each extracted chunk of a sentence is a structure. For this, token embeddings of a single chunk's constituents are averaged. For the short text retrieval task, these chunk representations again are averaged to obtain a single vector representation for the sentence.

`chunk-NS`: same as `chunk`, w/o stop-words.

`dep`: dependency relations are encoded as a combined vector of its head and tail word. Three

strategies are tested to encode dependency relations as vectors *a*) both vectors are concatenated (`-concat`) *b*) both vectors are averaged (`-avg`) *c*) for each word, we concatenate the word vector itself with the averaged vectors of its dependents (`-depavg`).

Figures 2 (c-e) show the structures for an example sentence. The following two baseline approaches produce a single vector representation for the entire sentence:

`CLS`: the artificial `[CLS]` token of BERT-based models, which is added to every sentence as a meta-token and which is frequently used as a vector representation for the entire sequence in downstream tasks;

`BoW`: all embeddings are averaged (bag-of-words).

4 Experiments

Several word-level- and sentence-level retrieval tasks of different granularity are tested. We also compare with static word embeddings provided by Mikolov et al. (2013, word2Vec)[4] since our linguistic structures enable the composition of meaning due to the use of multiple tokens for a single structured representation. We investigate the retrieval performance using precision at k (P@k, $k = 1$ and $k = 5$) and mean average precision (mAP) and refer to the static word2Vec embedding as w2v and to the contextualized RoBERTa embedding as RB. To perform the retrieval evaluation based on gold standard data, we use labeled datasets, which means each word or sentence is labeled with one specific target class. We use the standard train and test splits for indexing and querying as indicated by each task-specific dataset.

We additionally run a simple classification benchmark test using the same datasets. As a classification approach, we opted to use a k-nearest neighbor (kNN) approach, which heavily relies on the retrieval performance and, thus, implicitly evaluates the retrieval performance. The kNN approach groups and counts the class labels of the top k retrieved training samples and uses the most prominent class label as a classification result. In case of ties, a random label of the most prominent class labels is chosen. Here, we report F_1 scores on the test sets and determine the hyper-parameter k by using the validation set of the respective task benchmarks.[5]

4.1 Word-level tasks

Named Entity Recognition (NER) We use NER as a coarse-grained task. We evaluate the retrieval performance on the two common English benchmark datasets *CoNLL-2003* (Tjong Kim Sang and De Meulder, 2003) and *OntoNotes Release 5.0* (Weischedel et al., 2013).[6]

For retrieval, we only use structures consisting of entity-labeled tokens, i.e. excluding the 'other' class — with the goal to find more structures having the same label as the query. For NER, searching for non-entities, and including their scores, would only increase the reported performance, because the majority of labels are actually 'other'.

Both word-level structures explained in Section 3.2 are tested. An issue arises when retrieving token spans instead of whole sentences because the unit of retrieval is some linguistic structure that does not necessarily map perfectly to an entity span. Since there is no proper solution to this issue, we validate the appropriateness of our linguistic structures used for retrieval via named-entity classification. The classification scores allow interpretation and connection to SOTA results, but we note that those results are only for anecdotal purposes and cannot be properly compared to SOTA systems because of the simplicity and different objective of our approach.

Word Sense Disambiguation (WSD) can be considered as a fine-grained multi-class problem with thousands of classes where each word sense is a class. We evaluate retrieval and classification performance on a wide range of WSD datasets. In particular, we use the following datasets provided by UFSAC (Vial et al., 2018)[7]: *SemCor* (Miller et al., 1993), *WordNet Gloss Tag*[8] (WNGT) consisting of all WordNet (Miller, 1995) definitions, SensEval 2 (Edmonds and Cotton, 2001) & 3 (Litkowski, 2004) as well as SemEval 2007 Task 7 (Navigli et al., 2007) & 17 (Pradhan et al., 2007). The *SemCor* and WNGT datasets are used as training corpora with SemEval 2007 Task 7 and 17 as query

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

[5] If an explicit validation set is not supplied, we split the original training set (80/20) and use a random subset for validation and the remainder for training.

[6] We apply the split proposed by Pradhan et al. (2013) for OntoNotes as there is no official dataset split.

[7] https://github.com/getalp/UFSAC

[8] https://wordnetcode.princeton.edu/glosstag.shtml

datasets. For SensEval 2 and 3, we use their respective training and test sets.

In analogy to NER, we only use words that need disambiguation as queries for the retrieval evaluation. Since WSD is mostly the task of disambiguating a single word, we only use the `token` structure.

4.2 Sentence Level Tasks

Short-text retrieval evaluates the performance of retrieving semantically similar sentences ideally labeled with the same class. This task can be seen as a binary text classification problem. First, we try to find more tweets containing offensive language given an offensive tweet from the OLID dataset (Zampieri et al., 2019)[9] provided by the OffensEval 2019 Shared Task. Second, we want to obtain more negative or positive tweets from the Twitter Airline sentiment dataset[10]. Our intuition is that some very specific parts of a sentence (comparable to a particular linguistic structure) are responsible for triggering a particular class, e.g. making a tweet sound either offensive or negative.

Relation Identification is a multi-class classification problem, where the label space contains between 10 and 19 classes. We use three standard benchmarks from the SemEval[11] challenges for relation classification: SE'07 (SemEval 2007; Girju et al., 2007), SE'10 (SemEval 2010; Hendrickx et al., 2010), and SE'18 (SemEval 2018; Gábor et al., 2018). SE'07 and SE'10 focus on the classification of semantic relations between pairs of nominals. E.g. 'tea' and 'ginseng' are in an `ENTITY-ORIGIN(`e_1`,`e_2`)` relationship in the sentence 'The cup contained tea from dried ginseng'. SE'18 focuses on domain-specific semantic relations from scientific articles and provides entire paragraphs instead of single sentences.

We apply the sentence-level templates mentioned in Section 3.1 and additionally apply a specifically designed template structure, which involves the path between two given entities in a dependency path. The dependency path as a feature has been proven to be beneficial for relation extraction in multiple previous works.We define the feature vector $\mathbf{x}$ to be the concatenation of vectors for each entity $\mathbf{e}_{\{1,2\}}$ and the path $\mathbf{p}$, where each individual vector is the average vector of the words included: $\mathbf{x} := \mathbf{e}_1 \oplus \mathbf{e}_2 \oplus \mathbf{p}$ (cf. Fig. 2f).

Frame Identification or classification is considered to be a fine-grained multi-class classification problem since every frame is its own class. We evaluate the performance on FrameNet (Baker et al., 1998). The latest release of the dataset is FrameNet-1.7, but FrameNet-1.5 is by far the most commonly used one in the literature. We report results for both versions. For this work, we only use the dataset of fulltext annotations which provides 78 documents for FrameNet-1.5 and 108 documents for FrameNet-1.7. To generate data splits for both versions, we use 23 documents to extract the test set following the previous work (Das et al., 2014; Peng et al., 2018) and 16 documents are used as development set (Hermann et al., 2014), whereas the remaining documents are used as training set. Each frame is associated with one or more frame evoking elements commonly referred to as `lexical-units`. For example, the frame 'Abandonment' can be evoked by the `lexical-units` 'abandon', 'depart' or 'leave'. To find sentences that represent the same frame, we use the following task-dependent structures in addition to the default structures:

`lexical-unit`: This structure is based on the target words and phrases corresponding to the `lexical-unit` of the respective frame. Unlike PropBank (Palmer et al., 2005), where the target predicate is always a verb, FrameNet contains ten different types of lexical units such as nouns, adjectives, and prepositions. Embeddings of multi-token lexical units are averaged.

`subj-v-obj`: This structure is based on the concatenation of `subject-verb-object` triples, which have demonstrated competitive performance for unsupervised semantic frame induction tasks (Ustalov et al., 2018). For non-verb lexical units with no subject and object, we just consider the lexical unit.

5 Results

For discussion, we focus on P@1 scores because we believe this is the most important metric for practical applicability. As expected, we observe significantly better performance using contextual word embeddings as compared to static word embeddings across all tasks. However, our goal is not to compare these two types of embeddings,

[9] https://competitions.codalab.org/competitions/20011

[10] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[11] https://semeval.github.io/

Data	Aggregation token	SPS	
CoNLL-2003 (w2v)	37.1	*38.9*	mAP1K
	71.3	*79.8*	P@1
	64.5	*70.3*	P@5
CoNLL-2003 (RB)	***48.0***	***48.0***	mAP1K
	87.3	87.2	P@1
	78.1	***79.3***	P@5
OntoNotes-v5 (w2v)	26.6	*29.6*	mAP1K
	49.7	*50.5*	P@1
	38.7	*44.9*	P@5
OntoNotes-v5 (RB)	***38.4***	36.0	mAP1K
	75.7	75.3	P@1
	64.4	***64.5***	P@5

Table 1: NER retrieval results. We use the mean average precision (mAP) estimate of the top 1K nearest neighbors.

Data	Embedding w2v	RB	
SensEval 2	45.9	***65.9***	mAP1K
	38.8	***75.1***	P@1
	40.4	***69.7***	P@5
SensEval 3	45.7	***64.2***	mAP1K
	40.5	***72.3***	P@1
	45.1	***68.7***	P@5
SemEval '07 T7 (SemCor)	35.6	***41.4***	mAP1K
	22.3	*27.8*	P@1
	22.5	*26.5*	P@5
SemEval '07 T7 (WNGT)	31.8	*38.6*	mAP1K
	25.0	***32.7***	P@1
	24.7	***29.9***	P@5
SemEval '07 T17 (SemCor)	50.0	***63.3***	mAP1K
	41.7	***62.6***	P@1
	42.7	***57.5***	P@5
SemEval '07 T17 (WNGT)	37.1	*53.0*	mAP1K
	32.4	*54.7*	P@1
	29.6	*44.5*	P@5

Table 2: WSD Retrieval results for the `token` structure.

but to evaluate if aggregation of embeddings along linguistically informed lexical structures provides benefits for retrieval compared to the baselines regardless of the type of embedding.

Named-entity recognition: Table 1 shows the retrieval results for the CoNLL-2003 and OntoNotes v5 datasets. The retrieval performances of the two structures differ depending on the type of word embedding, we can see a rough increase of 10-15% for each dataset and aggregation strategy. With static word embeddings, the SPS structure shows improved performance compared to the token structure. A likely explanation is that averaging vectors of neighboring words inherently creates a kind of composite embedding that is unique for the combination of words. This is supported by the observation that for CWEs, there is only a minor difference between both linguistic structures. For small k, `SPS` is marginally better while `token` outperforms `SPS` on the mAP1K metric on the OntoNotes dataset.

The classification results for CoNLL-2003 and

Data	Aggregation CLS	BoW	token	word	word-NS	chunk	chunk-NS	dep-cat	dep-avg	dep-depavg	
Twitter-	-	*75.9*	63.0	64.3	64.3	66.2	65.6	65.4	66.0	65.6	mAP
Airline	-	*85.6*	71.9	27.4	71.9	56.4	70.6	62.4	59.1	65.2	P@1
(w2v)	-	*86.2*	58.6	51.9	59.5	62.0	57.9	62.6	59.2	62.3	P@5
Twitter-	73.7	***79.0***	63.8	64.7	65.7	78.8	77.9	63.4	65.1	64.3	mAP
Airline	23.5	88.9	74.4	77.5	81.2	***90.0***	89.0	67.8	71.3	68.4	P@1
(RB)	35.3	88.3	72.7	75.8	79.3	***89.7***	88.2	67.1	68.5	67.2	P@5
Offens-	-	39.3	47.5	48.4	***51.5***	47.9	49.2	45.4	45.7	46.0	mAP
Eval'19	-	52.5	60.4	66.2	*69.2*	60.8	62.5	57.5	56.2	60.0	P@1
(w2v)	-	46.8	61.2	64.5	*66.4*	62.4	63.5	58.7	56.8	60.8	P@5
Offens-	29.6	39.4	43.1	43.2	*44.7*	40.2	40.2	41.8	39.9	43.8	mAP
Eval'19	62.5	49.2	67.5	66.2	***70.4***	48.8	48.3	59.2	63.3	63.3	P@1
(RB)	56.7	47.6	66.8	66.3	***68.8***	52.3	50.9	62.3	56.8	63.5	P@5

Table 3: Short text retrieval results.

OntoNotes-v5 are shown in Table 6[12]. Overall, the picture is very similar to retrieval. There is only a minor difference between both structures when using contextual embeddings. While the classification with the k-NN approach does not reach SOTA performance, the scores show that both linguistic structures are generally useful to retrieve named entities of the same type.

Word sense disambiguation: Table 2 shows the WSD retrieval results for the various pairs of query and background datasets. For SemEval '07 scores for task 17 are considerably higher as it is not as fine-grained as task 7. Furthermore, the use of SemCor as a background corpus is superior to WNGT. These dataset characteristics are independent of the choice of word embedding type.

The performance of k-NN classification with static word embeddings is always close to the most frequent sense (MFS) baseline (cf. Tab. 7 in the appendix). With CWEs, however, this baseline is beaten by a large margin (cf. Tab. 6).

Short-text retrieval: Table 3 shows the retrieval results for tweet labels. Aggregating embeddings with the `chunk` structure improves the retrieval performs best for sentiment analysis (90% for TwitterAirline and RB). For offensive language, the `word-NS` strategy performs best (70.4% for OffensEval'19 and RB). The reason for this could be that longer phrases are required to express a sentiment but a single word is enough to express offensive content. It is thus highly category-dependent which strategy to use for semantic retrieval.

Relation identification: A common pattern for all datasets is that simple linguistic structures perform worse in terms of P@1 than the baseline BoW approach (cf. Tab. 4). Among the simple linguis-

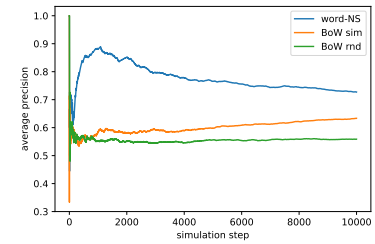
[12]Complete results can be found in Tables 7 and 8 in the appendix.

Data		CLS	BoW	token	word	word-NS	chunk	chunk-NS	dep-cat	dep-avg	dep-depavg	dep-path	
		Aggregation											
SE'18 (w2v)	-		32.9	31.1	30.4	31.2	31.0	30.9	30.6	30.8	31.2	***36.8***	mAP
	-		39.1	33.1	29.7	30.3	30.6	31.4	25.7	27.7	31.7	*46.0*	P@1
	-		34.6	30.9	31.7	33.4	31.4	31.3	27.1	30.2	31.1	*43.3*	P@5
SE'18 (RB)		31.9	34.5	32.1	31.4	32.0	32.1	32.2	31.8	32.2	32.4	*35.3*	mAP
		35.4	40.3	29.7	32.0	34.9	37.7	32.9	33.7	32.9	34.9	***52.9***	P@1
		34.6	37.8	33.5	32.2	35.0	33.4	34.9	35.0	33.4	34.1	***46.9***	P@5
SE'10 (w2v)	-		12.7	9.0	9.5	9.8	10.8	10.6	11.1	11.3	11.4	*22.2*	mAP
	-		35.5	9.9	14.4	15.6	21.9	21.8	22.7	22.5	23.0	*58.6*	P@1
	-		30.3	10.0	11.3	14.6	19.3	19.2	19.7	19.9	20.4	*50.0*	P@5
SE'10 (RB)		10.3	14.1	11.5	12.6	12.3	12.8	13.2	15.1	13.5	15.5	***26.5***	mAP
		31.6	40.6	26.0	26.8	27.3	32.0	32.4	38.3	27.5	37.6	***73.0***	P@1
		27.0	35.9	22.0	23.3	23.5	28.6	29.0	34.0	26.3	33.4	***66.5***	P@5
SE'07 (w2v)	-		32.2	29.2	29.6	29.8	30.5	30.4	30.5	30.6	30.8	***37.9***	mAP
	-		39.2	17.9	15.1	32.8	31.9	33.2	37.0	35.2	34.8	*53.6*	P@1
	-		36.5	20.2	22.9	30.2	32.2	32.4	31.8	32.6	33.3	*49.3*	P@5
SE'07 (RB)		30.8	31.6	30.6	31.2	31.5	31.1	31.3	32.2	31.3	32.5	*37.0*	mAP
		36.8	39.9	36.2	37.7	40.4	40.8	39.5	43.2	34.8	43.7	***61.9***	P@1
		33.7	37.3	32.9	34.9	35.6	37.2	35.3	39.9	34.3	38.6	***53.8***	P@5

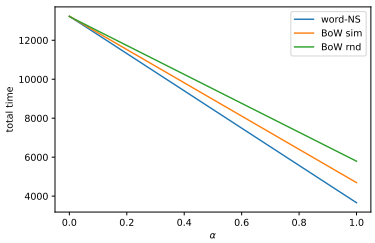
Table 4: Relation identification retrieval results.

Data		CLS	BoW	token	word	word-NS	chunk	chunk-NS	dep-cat	dep-avg	dep-depavg	lexical-unit	subj-v-obj	
		Aggregation												
FN1.5 (w2v)	-		1.8	0.9	1.0	1.2	1.6	1.5	1.5	1.5	1.4	***45.1***	41.7	mAP
	-		3.2	0.6	0.8	1.4	2.1	1.8	1.2	1.7	1.9	*80.3*	70.2	P@1
	-		3.3	0.7	0.9	1.3	2.0	1.5	1.5	1.7	1.8	*73.4*	66.8	P@5
FN1.5 (RB)		1.2	1.6	1.3	1.3	1.4	1.6	1.6	1.7	1.4	1.5	*38.0*	31.1	mAP
		1.6	2.2	1.8	2.2	1.8	2.4	2.5	2.7	2.0	2.3	***83.4***	77.0	P@1
		1.8	2.5	1.7	2.1	2.2	2.6	2.5	2.7	2.3	2.5	***74.2***	67.1	P@5
FN1.7 (w2v)	-		1.7	0.8	0.9	1.1	1.4	1.4	1.5	1.4	1.3	***44.6***	41.4	mAP
	-		3.5	0.9	0.8	1.4	2.3	1.6	1.2	1.4	1.5	*79.3*	70.6	P@1
	-		3.4	0.8	0.7	1.2	1.8	1.6	1.5	1.5	1.5	*74.7*	67.5	P@5
FN1.7 (RB)		1.1	1.5	1.2	1.3	1.4	1.5	1.5	1.6	1.3	1.5	*37.8*	30.8	mAP
		1.7	2.7	1.7	2.0	2.4	2.6	2.8	2.8	2.1	2.5	***84.0***	77.1	P@1
		1.8	2.4	1.8	1.9	2.2	2.5	2.4	2.7	2.2	2.6	***75.5***	68.2	P@5

Table 5: Frame identification retrieval results.



(a)



(b)

Figure 3: Simulation of similarity-based data labeling for offensive tweets: average agreement of subsequent sample labels (a), simulated label cost reduction depending on relative time saving due to reduced cognitive shifting (b).

tic structures, the `dependency-depavg` still performs consistently better than other structures, probably because it covers more words than others. `BoW` also consistently produces better results than the `CLS` approach, which questions the practical usability of the `[CLS]` meta-token for downstream tasks. The specialized `dependency-path` structure, however, improves the results by a large margin, almost doubling the `BoW` results and even tripling the `token`-based results (cf. e.g. 73% P@1 for SE'10 and RB). We believe that `BoW` and `dependency-path` work so well because relations require even more content than sentiments and `dependency-path` focuses the content on the important part of the sentence.

Frame identification: Table 5 shows the retrieval results for frame identification. The `lexical-unit` structure has shown the best performance (∼84% P@1 for RB), followed by `subj-v-obj` (∼77% P@1 for RB). All other simple sentence-level structures perform significantly worse. In FrameNet, one sentence can have multiple lexical units which invoke different frames. Simple structures do not capture this and treat each structure as a representative for the whole sentence. The performance is further negatively affected by the very large number of classes in FrameNet $(1,000+)$ in comparison to other tasks discussed in this work. Thus, high precision, i.e. one representative embedding laying out only the frame evoking lexical unit suppresses the noise that other structures introduce.

6 Application

Based on our findings, we investigate two downstream applications. First, similarity-based retrieval improved with linguistic information can be used to speed up manual labeling of text data. Second, aggregated CWEs can be used for rapid nearest neighbor classification with small training data.

Data labeling: Utilizing similarity information during annotation tasks can reduce annotation time and costs. In neuroscience, task switching is a well-studied phenomenon describing prolonged cognitive processing times due to altered tasks and task parameters (Rogers and Monsell, 1995). Vice versa, tasks can be solved faster in a series if parameters stay similar. This circumstance can be used to improve data labeling processes by presenting more similar instead of random samples to human annotators. We simulate the potential gains of such a process for selected aggregation strategies.

For this, we assume that labeling a single random example s_i takes the maximum amount of one time unit t. Labeling of the next most similar sample reduces cognitive processing time to $t - \alpha \times t \times sim(s_i, s_{i+1}) \times \beta$ with regard to the similarity of the two samples and a task-dependent parameter α representing its complexity, i.e. the upper bound of potential speed-up relative to t. Speed-up is expected if the labels of s_i and s_{i+1} agree, in this case setting $\beta = 1$, and $\beta = 0$ otherwise. Figure 3 shows the result of such a simulation on the OLID dataset. Similarity-based retrieval of samples for labeling achieves higher agreement between consecutive labels than random sample selection (cf. Fig. 3a). The best performing strategy `word-NS` outperforms `BoW`, especially in the early steps of the simulation. Figure 3b shows that significant time savings can be expected. For $\alpha = .4$, an assumed upper bound of 40% reduction of cognitive processing time per sample, for instance, the simulation shows a total time saving of ca. 10 %.

Rapid nearest neighbor classification: Table 6 shows a summary of kNN classification experiments with the best performing setup for each task and dataset, which was identified using a held-out validation set and evaluated on the held-out test set. Interestingly, the best classification setups do not correlate with the precision at k scores in the retrieval setup, but rather the mAP scores. While the classification results do not reach SOTA, they still achieve considerable results over a standard baseline. Much shorter training and prediction times of kNN-classification compared to fine-tuning transformers make it an appealing approach in some scenarios despite the lower performance.

Furthermore, kNN can be used in few-shot classification scenarios. We test the performance of the classifier with increasing dataset size, where we randomly select training sentences for indexing. Results are plotted in Figure 4. For the word-level task of NER (Fig. 4a), we can see that as few as $3,000$ sentences are sufficient to reach a decent performance that only slightly increases with more training data. The findings for the sentence-level tasks (Fig. 4b) are even more drastic, where, depending on the task and the available training data, as few as 300 to $1,000$ sentences are sufficient to reach a similar performance as compared to using the entire training data.

Data	Embedding	Aggregation	k	F1
CoNLL-2003	RB	`SPS`	1	79.6
OntoNotes-v5	RB	`SPS`	9	65.9
SensEval 2	RB	`token`	8	78.1
SensEval 3	RB	`token`	15	73.3
SemEval '07 T7 (WNGT)	RB	`token`	1	69.7
SemEval '07 T17 (SemCor)	RB	`token`	7	63.6
TwitterAirline	RB	`BoW`	29	87.8
OffensEval'19	RB	`word-NS`	75	63.6
SE'07	w2v	`dep-path`	1	43.4
SE'10	RB	`dep-path`	5	78.7
SE'18	RB	`dep-path`	5	35.9
FN1.5	RB	`lexical-unit`	1	63.9
FN1.7	RB	`lexical-unit`	1	61.9

Table 6: Classification results using kNN for word-level tasks (upper part) and sentence-level tasks (lower part). k refers to the best identified validation k.

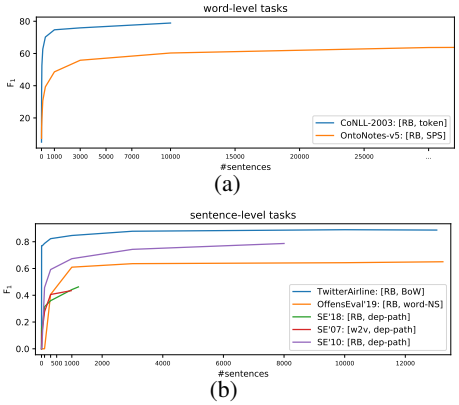


Figure 4: kNN performance for increasing training dataset sizes for the word-level task of NER (a) and the sentence-level tasks of short text classification and relation classification (b).

7 Conclusion

We presented an analysis of different linguistically informed aggregation strategies for word embeddings in an information retrieval setting to find semantic units of the same class for different NLP tasks. Our experiments show that more fine-grained label sets perform better with specifically designed task-dependent linguistic structures, whereas coarse-grained tasks such as short-text classification, work quite well with simple structures such as `chunk`, `word-NS`, or even the `BoW` baseline. We believe that particularly for the short-text classification tasks, certain keywords often are sufficient to trigger a certain class (e.g. offensive words). This can also be observed for word-level tasks. It is thus highly dependent on the task at hand if explicit structures based on external linguistic knowledge can be beneficial. We showed that more complex tasks benefit from both, linguistic structures and contextualized word embeddings. We also showed that for simple k nearest neigh-

bor classification, only a certain amount of training data is sufficient to reach a decent performance. Use cases of this work include support for rapid training data collection, manual coding/annotation of datasets e.g. in social science and humanities applications, retrieval of similar language use in eDiscovery tasks, and many more.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, NM, USA.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, ACL '98, pages 86–90, Montréal, QC, Canada.

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: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

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: Human Language Technologies*, pages 4171–4186, Minneapolis, MN, USA.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, LO, USA.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 13–18, Prague, Czech Republic.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1458, Baltimore, MD, USA.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in BERT track syntactic dependencies? In *Natural Language, Dialog and Speech (NDS) Symposium*, pages 1–7, New York, NY, USA.

Kenneth C. Litkowski. 2004. Senseval-3 task: Word sense disambiguation of WordNet glosses. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 13–16, Barcelona, Spain.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, AZ, USA.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the Association of Computing Machinery (ACM)*, 38(11):39–41.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of a Human Language Technology Workshop.*, pages 303–308, Plainsboro, NJ, USA.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. SemEval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, Prague, Czech Republic.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1492–1502, New Orleans, LA, USA.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, New Orleans, LA, USA.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong.

Robert D. Rogers and Stephen Monsell. 1995. Costs of a predictible switch between simple cognitive tasks. *Journal of Experimental Psychology*, 124(2):207–231.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning*, page 142–147, Edmonton, Canada.

Dmitry Ustalov, Alexander Panchenko, Andrei Kutuzov, Chris Biemann, and Simone Paolo Ponzetto. 2018. Unsupervised Semantic Frame Induction using Triclustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 55–62, Melbourne, VIC, Australia.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2018. UFSAC: Unification of Sense Annotated Corpora and Tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. StructBERT: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. OntoNotes Release 5.0 LDC2013T19. *Philadelphia: Linguistic Data Consortium*.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*, pages 161–170, Erlangen, Germany.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, MN, USA.

A KNN Results

Masking + Embedding	CoNLL-2003	OntoNotes-v5	SensEval 2	SensEval 3	SemEval '07 T7 (SemCor)	SemEval '07 T7 (WNGT)	SemEval '07 T17 (SemCor)	SemEval '07 T17 (WNGT)	
			Data						
MFS	-	-	55.3	54.4	63.60	58.0	51.8	38.9	F1
`token` (w2v)	3	25	25	24	24	8	20	25	k
	64.5	44.9	54.8	51.8	62.9	58.5	50.7	43.9	F1
`token` (RB)	1	11	8	15	6	1	7	1	k
	79.4	65.6	78.1	73.3	69.6	69.7	63.6	60.7	F1
`SPS` (w2v)	3	16	-	-	-	-	-	-	k
	73.5	52.3	-	-	-	-	-	-	F1
`SPS` (RB)	3	9	-	-	-	-	-	-	k
	79.6	65.9	-	-	-	-	-	-	F1

Table 7: Word-level classification results using KNN. Showing the best identified hyperparameter k and the F1 score.

Data	`CLS`	`BoW`	`token`	`word`	`word-NS`	`chunk`	`chunk-NS`	`dep-cat`	`dep-avg`	`dep-depavg`	`dep-path`	`lexical-unit`	`subj-v-object`	
							Masking							
Twitter-Airline	-	9	22	5	133	113	31	194	200	194	-	-	-	k
(w2v)	-	83.3	62.8	68.0	75.3	77.0	73.7	78.9	79.9	78.5	-	-	-	F1
Twitter-Airline	42	29	24	17	14	58	21	89	141	29	-	-	-	k
(RB)	83.9	87.8	81.7	82.5	83.5	81.8	82.3	79.9	81.5	79.3	-	-	-	F1
Offens-Eval'19	-	16	160	180	164	154	84	30	67	39	-	-	-	k
w2v	-	42.2	56.4	59.5	59.3	56.9	55.1	51.3	56.0	54.4	-	-	-	F1
Offens-Eval'19	8	54	54	44	75	66	51	38	52	27	-	-	-	k
(RB)	33.4	46.1	61.7	60.0	63.6	56.9	60.0	56.7	61.1	55.5	-	-	-	F1
SE'18	-	6	4	13	4	26	6	8	2	9	3	-	-	k
(w2v)	-	27.9	15.8	14.4	16.5	15.6	22.4	19.8	15.7	19.7	27.9	-	-	F1
SE'18	10	4	14	15	18	38	27	25	2	20	5	-	-	k
(RB)	21.5	26.6	21.6	17.3	20.2	15.2	16.5	24.1	25.0	17.6	35.9	-	-	F1
SE'10	-	65	11	24	16	41	30	23	24	22	107	-	-	k
(w2v)	-	40.7	9.6	11.3	17.2	22.7	22.4	24.4	27.8	24.4	67.0	-	-	F1
SE'10	14	17	90	28	28	42	49	38	15	9	5	-	-	k
(RB)	33.9	50.5	24.8	29.0	30.0	34.8	34.1	31.2	40.2	41.7	78.7	-	-	F1
SE'07	-	1	16	6	10	2	7	2	1	16	1	-	-	k
(w2v)	-	22.3	8.6	7.0	8.6	12.6	12.6	14.7	16.5	11.3	43.4	-	-	F1
SE'07	6	2	5	4	10	3	3	11	3	1	12	-	-	k
(RB)	11.0	22.4	13.4	13.4	10.8	18.8	14.4	17.2	23.0	26.5	41.6	-	-	F1
FrameNet-1.5	-	24	42	44	41	79	92	79	27	25	-	1	1	k
(w2v)	-	1.5	0.2	0.4	1.1	2.6	1.7	1.8	2.1	1.9	-	58.8	55.1	F1
FrameNet-1.5	20	14	44	34	49	26	66	60	17	24	-	1	1	k
(RB)	0.8	1.4	1.6	2.2	2.9	3.1	3.0	1.8	2.9	3.4	-	63.9	54.3	F1
FrameNet-1.7	-	9	2	62	60	64	61	41	113	38	-	1	1	k
(w2v)	-	1.3	0.4	0.8	1.1	2.5	1.9	1.7	2.2	2.0	-	56.3	52.6	F1
FrameNet-1.7	2	13	76	38	36	49	65	83	41	69	-	1	1	k
(RB)	0.6	1.7	1.5	2.2	3.1	3.4	3.3	1.8	3.1	3.0	-	61.9	53.1	F1

Table 8: Sentence-level classification results using KNN. Showing the best identified hyperparameter k and the F1 score.