Building a Web-Scale Dependency-Parsed Corpus from Common Crawl

Alexander Panchenko[‡], Eugen Ruppert[‡], Stefano Faralli[†], Simone P. Ponzetto[†], Chris Biemann[‡]
‡ University of Hamburg, Department of Informatics, Language Technology Group, Germany
† University of Mannheim, School of Business Informatics and Mathematics, Data and Web Science Group, Germany
{panchenko,ruppert,biemann}@informatik.uni-hamburg.de
{stefano,simone}@informatik.uni-mannheim.de

Abstract

We present DEPCC, the largest-to-date linguistically analyzed corpus in English including 365 million documents, composed of 252 billion tokens and 7.5 billion of named entity occurrences in 14.3 billion sentences from a web-scale crawl of the COMMON CRAWL project. The sentences are processed with a dependency parser and with a named entity tagger and contain provenance information, enabling various applications ranging from training syntax-based word embeddings to open information extraction and question answering. We built an index of all sentences and their linguistic meta-data enabling quick search across the corpus. We demonstrate the utility of this corpus on the verb similarity task by showing that a distributional model trained on our corpus yields better results than models trained on smaller corpora, like Wikipedia. This distributional model outperforms the state of art models of verb similarity trained on smaller corpora on the SimVerb3500 dataset.

Keywords: text corpus, Web as a corpus, Common Crawl, dependency parsing, verb similarity, distributional semantics

1. Introduction

Large corpora are essential for the modern data-driven approaches to natural language processing (NLP), especially for unsupervised methods, such as word embeddings (Mikolov et al., 2013) or open information extraction (Banko et al., 2007) due to the "unreasonable effectiveness of big data" (Halevy et al., 2009). However, the size of commonly used text collections in the NLP community, such as BNC[1] or Wikipedia is in the range 0.1–3 billion tokens, which potentially limits coverage and performance of the developed models. To overcome this limitation, larger corpora can be composed of books, e.g. in (Goldberg and Orwant, 2013) a dataset of syntactic n-grams[2] was built from the 345 billion token corpus of the Google Books project.[3] However, access to books is often restricted, which limits use-cases of book-derived datasets. Another source of large amounts of texts is the Web. Multiple researchers investigated the use of the Web for construction of text corpora, producing resources, such as PUKWAC (Baroni et al., 2009) (2 billion of tokens) and ENCOW16 (Schäfer, 2015) (17 billion of tokens), yet the size of these corpora is still at least one order of magnitude smaller than the web-scale crawls, e.g. CLUEWEB[4] and COMMON CRAWL[5]. On the other hand, directly using the web crawl dumps is problematic for researchers as: (1) the documents are not preprocessed, containing irrelevant information, e.g. HTML markup; (2) big data infrastructure and skills are required; (3) (near)duplicates of pages disbalance the corpus; (4) documents are not linguistically analyzed, thus only shallow models can be used. The mentioned factors substantially limit the use of web-scale corpora in natural language processing research and applications.

The objective of this work is to address these issues and *make access to web-scale corpora a commodity* by providing a web-scale corpus that is ready for NLP experiments as it is linguistically analyzed and cleansed from noisy irrelevant content. Namely, in this paper, we present a technology for constructing of linguistically analyzed corpora from the Web and release DEPCC, the largest-to-date dependency-parsed corpus of English texts.

The COMMON CRAWL project regularly produces web-scale crawls featuring a substantial fraction of all public web pages. For instance, as of October 2017, the estimated number of pages on the Web is 47 billion[6], while the corresponding crawl contains over 3 billion pages. To put this number into perspective, according to the same source, the indexed Web contains about 5 billion pages.

COMMON CRAWL provides the data in the Web ARChive (WARC) format. Each crawl is provided in the raw form features full HTML pages with metadata or in the form of preprocessed archives containing texts (WET). The WET archives contain extracted plaintext from the raw crawls. For instance, the 29.5 Tb raw crawl archive (cf. Table 2) has a corresponding 4.8 Tb WET version with texts. The preprocessing used for producing the WET archives is limited to removal of HTML tags. After a manual check, we also noticed that in WET archives (1) some documents still contain HTML markup; (2) the archives contain document duplicates; (3) documents are written in various languages making it difficult to train language-specific linguistic models. Finally, most importantly, the WET dumps are not linguistically analyzed, which significantly limits their utility for language processing applications.

In this work, we address these limitations by constructing a text corpus from COMMON CRAWL, which is filtered from irrelevant and duplicate documents and is linguistically an-

[1] `http://www.natcorp.ox.ac.uk`
[2] `http://commondatastorage.googleapis.com/books/syntactic-ngrams/index.html`
[3] `https://books.google.com`
[4] `http://lemurproject.org/clueweb12`
[5] `http://www.commoncrawl.org`
[6] `http://www.worldwidewebsize.com` at 02.10.2017

	WaCkypedia	Wikipedia	PukWaC	GigaWord	ENCOW16	ClueWeb12	Syn.Ngrams	DEPCC
Tokens (billions)	0.80	2.90	1.91	1.76	16.82	N/A	345.00	251.92
Documents (millions)	1.10	5.47	5.69	4.11	9.22	733.02	3.50	364.80
Type	Encyclop.	Encyclop.	Web	News	Web	Web	Books	Web
Source texts	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Preprocessing	Yes	No	Yes	No	Yes	No	No	Yes
NER	No	No	No	No	Yes	No	No	Yes
Dependency-parsed	Yes	No	Yes	No	Yes	No	Yes	Yes

Table 1: Comparison of existing large text corpora for English with the DEPCC corpus.

alyzed. Namely, the contributions of this paper are the following:

1. We present a *methodology* for the creation of the text corpus from the web-scale crawls of COMMON CRAWL.
2. We present a *software* implementing the methodology in a scalable way using the MapReduce framework.
3. We present the largest-to-date *dependency parsed corpus* of English texts obtained using the developed methodology, also featuring *named entity tags*.
4. We show the utility of the web-scale corpora on the *verb similarity* task by outperforming the state of the art on the SimVerb3500 dataset (Gerz et al., 2016).

The corpus and the software tools are available online.[7] Namely, the corpus can be directly used without the need to download it on the Amazon S3 distributed file system, cf. Section 3.7.[8] The software tools used to build the corpus are distributed under an open source license. The terms of use of the corpus are described in Section 4.

2. Related Work

2.1. Large Scale Text Collections

In Table 1 we compare the DEPCC corpus to seven existing large-scale English corpora, described below.

WACKYPEDIA (Baroni et al., 2009) is a parsed version of English Wikipedia as of 2009. The articles are part-of-speech tagged with the TreeTagger (Schmid, 1994) and dependency parsed with the Malt parser (Nivre et al., 2007). Similarly to our corpus, the results are presented in the CoNLL format.[9] The 2017 version of WIKIPEDIA contains three times more tokens, compared to the version of 2009, yet the there are no distributions of linguistically analyzed recent dumps. PUKWAC is a dependency-parsed version of the UKWAC corpus (Baroni et al., 2009), which is processed in the same way as the WACKYPEDIA corpus.

GIGAWORD (Parker et al., 2011) is a large corpus of newswire, which is not dependency parsed. The CLUEWEB12 is a corpus similar to the raw COMMON CRAWL corpus: it contains archives of linguistically unprocessed web pages.

[7]https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/depcc.html

[8]https://commoncrawl.s3.amazonaws.com/contrib/depcc/CC-MAIN-2016-07/index.html

[9]http://www.universaldependencies.org/format.html

Stage of the Processing	Size (.gz)
Input raw web crawl (HTML, WARC)	29,539.4 Gb
Preprocessed corpus (simple HTML)	832.0 Gb
Preprocessed corpus English (simple HTML)	683.4 Gb
Dependency-parsed English corpus (CoNLL)	2,624.6 Gb

Table 2: Various stages of development of the corpus based on the COMMON CRAWL 2016-07 web crawl dump.

The authors of the GOOGLE SYNTACTIC NGRAMS corpus (Goldberg and Orwant, 2013) parsed a huge collection of books and released a dataset of syntactic dependencies. However, the source texts are not shared due to copyright restrictions, which limits potential use-cases of this resource.

Finally, ENCOW16 (Schäfer, 2015) is a large-scale web corpus, which is arguably the most similar one to DEPCC. The authors also rely on the Malt parser and perform named entity tagging. However, this corpus contains roughly 15 times less tokens than DEPCC.

2.2. Common Crawl as a Corpus

Kolias et al. (2014) present an exploratory study of one of the early versions of the COMMON CRAWL. The authors provide various descriptive statistics of the dataset regarding language distribution, formats of the documents, etc.

COMMON CRAWL was used construct a large-scale Finnish Parsebank consisting of 1.5 billion tokens in 116 million sentences (Laippala and Ginter, 2014). The texts were morphologically and syntactically analyzed. In addition, distributional vector space representations of the words were obtained using the word2vec toolkit (Mikolov et al., 2013). The resources were made available under an open license.

GloVe (Pennington et al., 2014) is an unsupervised model for learning distributional word representations similar to word2vec. The authors distribute[10] two models trained on the English part of a COMMON CRAWL corpus (comprising respectively 42 and 820 billion of tokens), which are often used to build neural NLP systems, such as (Tsuboi, 2014). The models were trained on the COMMON CRAWL documents texts tokenized with the Stanford tokenizer. In addition, the smaller training corpus was lowercased.

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

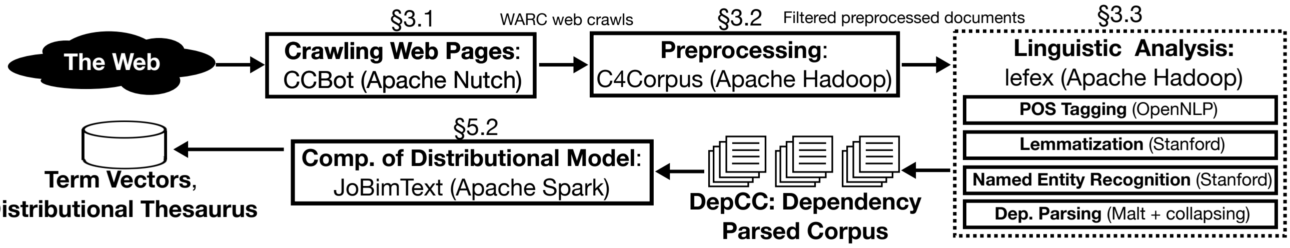

Figure 1: Outline of the corpus construction approach and experiments described in the paper.

3. Building a Web-Scale Dependency-Parsed Corpus in English from Common Crawl

Figure 1 shows how a linguistically analyzed corpus is built from the Web. First, web pages are downloaded by the web crawler of COMMON CRAWL, called CCBot. Second, preprocessing, involving elimination of duplicates and language detection, is performed using the C4Corpus tool. Finally, we perform linguistic analysis of the corpus and save the results in the CoNLL format (cf. Section 3.4.).

3.1. Input Web Crawl: the Common Crawl

The DEPCC corpus is based on the crawl of February 2016[11] containing more than 1.73 billion URLs. The COMMON CRAWL URL index for this crawl is available online[12], while the original files are located in the "commoncrawl" bucket on the S3 distributed file system.[13] As summarized in Table 2, the total size of the compressed HTML WARC files is about 30 Tb.

3.2. Preprocessing of Texts: the C4Corpus Tool

The raw corpus was processed with the C4Corpus tool (Habernal et al., 2016) and is available on the distributed cloud-based file system Amazon S3.[14] The tool performs preprocessing of the raw corpus, in five phases:

1. Language detection, license detection, and removal of boilerplate page elements, such as menus;
2. "Exact match" document de-duplication;
3. Detecting near duplicate documents;
4. Removing near duplicate documents;
5. Grouping the final corpus by language and license.

The resulting output is a gzip-compressed corpus with a total size of 0.83 Tb (cf. Table 2). For further processing, we selected only English texts with the total size of 0.68 Tb, based on the language detection in the first phase. Note that we use all texts written in English, not only those published under the CC-BY license.

[11] http://commoncrawl.org/2016/02/

[12] http://index.commoncrawl.org/CC-MAIN-2016-07

[13] s3://commoncrawl/crawl-data/CC-MAIN-2016-07

[14] s3://commoncrawl/contrib/c4corpus/CC-MAIN-2016-07

3.3. Linguistic Analysis of Texts

Linguistic analysis consists of four stages presented in Figure 1 and is implemented using the Apache Hadoop framework[15] for parallelization and the Apache UIMA framework[16] for integration of linguistic analysers via the DKPro Core library (Eckart de Castilho and Gurevych, 2014).[17]

3.3.1. POS Tagging and Lemmatization

For morphological analysis of texts, we used OpenNLP part-of-speech tagger and Stanford lemmatizer.

3.3.2. Named Entity Recognition

To detect occurrences of persons, locations, and organizations we use the Stanford NER tool (Finkel et al., 2005).[18] Overall, 7.48 billion occurrences of named entities were identified in the 251.92 billion tokens output corpus.

3.3.3. Dependency Parsing

To make large-scale parsing of texts possible, a parser needs to be not only reasonably accurate but also fast. Unfortunately, the most accurate parsers, such as Stanford parser based on the PCFG grammar (De Marneffe et al., 2006), according to our experiments, take up to 60 minutes to process 1 Mb of text on a single core, which was prohibitively slow for our use-case (details of the hardware configuration are available in Section 3.5.). We tested all versions of the Stanford, Malt (Hall et al., 2010), and Mate (Ballesteros and Bohnet, 2014) parsers for English available via the DKPro Core framework. To dependency-parse texts, we selected the Malt parser, due to an optimal ratio of efficiency and effectiveness (parsing of 1 Mb of text per core in 1–4 minutes). This parser was successfully used in the past for the construction of linguistically analyzed web corpora, such as PUKWAC (Baroni et al., 2009) and ENCOW16 (Schäfer, 2015). While more accurate parsers exist, e.g. the Stanford parser, according to our experiments, even the neural-based version of this parser is substantially slower. On the other hand, as shown by Chen and Manning (2014), the performance of the Malt parser is only about 1.5–2.5 points below the neural-based Stanford parser. In particular, we used the stack model based on the projective transition system with the Malt.[19]

[15] https://hadoop.apache.org

[16] https://uima.apache.org

[17] https://github.com/uhh-lt/lefex

[18] stanfordnlp-model-ner-en-all.3class.distsim.crf, 20.04.2015

[19] The used model is de.tudarmstadt.ukp.dkpro.core.maltparser-upstream-parser-en-linear, version 20120312.

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	NER
# newdoc url = `http://www.poweredbyosteons.org/2012/01/brief-history-of-bioarchaeological.html`									
# newdoc s3 = `s3://aws-publicdatasets/common-crawl/crawl-data/CC-MAIN-2016-07/segments...`									
...									
# sent_id = `http://www.poweredbyosteons.org/2012/01/brief-history-of-bioarchaeological.html#60`									
# text = The American Museum of Natural History was established in New York in 1869.									
0	The	the	DT	DT	_	2	det	2:det	O
1	American	American	NNP	NNP	_	2	nn	2:nn	B-Organization
2	Museum	Museum	NNP	NNP	_	7	nsubjpass	7:nsubjpass	I-Organization
3	of	of	IN	IN	_	2	prep	_	I-Organization
4	Natural	Natural	NNP	NNP	_	5	nn	5:nn	I-Organization
5	History	History	NNP	NNP	_	3	pobj	2:prep_of	I-Organization
6	was	be	VBD	VBD	_	7	auxpass	7:auxpass	O
7	established	establish	VBN	VBN	_	7	ROOT	7:ROOT	O
8	in	in	IN	IN	_	7	prep	_	O
9	New	New	NNP	NNP	_	10	nn	10:nn	B-Location
10	York	York	NNP	NNP	_	8	pobj	7:prep_in	I-Location
11	in	in	IN	IN	_	7	prep	_	O
12	1869	1869	CD	CD	_	11	pobj	7:prep_in	O
13	.	.	.	.	_	7	punct	7:punct	O
...									

Table 3: An excerpt from an output document in the CoNLL format: a document header plus a sentence are shown. Here, "ID" is a word index, "FORM" is word form, "LEMMA" is lemma or stem of word form, "UPOSTAG" is universal part-of-speech tag, "XPOSTAG" is language-specific part-of-speech tag, "FEATS" is a list of morphological features, "HEAD" is head of the current word, which is either a value of ID or zero, "DEPREL" is universal dependency relation to the "HEAD", "DEPS" is enhanced dependency graph in the form of head-deprel pairs, and "NER" is named entity tag.

The text downloaded from the Web has highly variable quality due to the inherent nature of user-generated content, but also unavoidable pre-processing errors, e.g. during the cleanup of incomplete HTML markup. To avoid crashes of the dependency parser caused by excessively long sentences, we filter all sentences longer than 50 tokens. Our manual analysis revealed that there are hardly any well-formed sentences of 50 tokens or more in this corpus.

3.3.4. Collapsing of Syntactic Dependencies

Collapsed and enhanced dependencies, such as the Stanford Dependencies (De Marneffe et al., 2006)[20] can be useful in various NLP tasks as they provide a more compact syntactic trees of a sentence, compared to the original dependency tree, thus reducing sparsity of syntax-aware representations.

To compensate the lack of the dependency enhancement in Malt, we use the system of (Ruppert et al., 2015)[21] to perform collapsing and enhancing of dependencies. The authors of the toolkit shown that (1) using the collapsed dependency representations substantially improves quality of construction of distributional thesauri based on sparse syntactic features; (2) the performance of the Stanford enhanced dependencies and the collapsed Malt dependencies on the same task are comparable. The advantage of using Malt with an external collapsing with respect to the Stanford parser, in our case, is speed.

Note that, both original and enhanced versions are saved respectively into the columns "DEPREL" and "DEPS" as illustrated in Table 3.

3.4. Format of the Output Documents

The documents are encoded in the CoNLL format as illustrated in Table 3. The corpus is released as a collection of 19,101 gzip-compressed files.

Each file is relatively small (around 150Mb) and is easy to download and work with locally during the development phase. However, to work with the entire corpus we recommend using some kind of parallelism, e.g. based on multiprocessing/multithreading or frameworks for distributed computing, such as Apache Hadoop/Spark/Flink.

3.5. Computational Settings

The linguistic analysis was performed on an Apache Hadoop 2.6 cluster using 341 containers each provided with one Intel Xeon CPU E5-2603v4@1.70GHz and 8Gb of RAM. The computational cluster consisted of 16 nodes plus a single head node. The job used 2.75 TB out of 2.82 TB available RAM and 356 out of 640 available Vcores.

3.6. Running Time

In total, the computations were completed in 110 hours in 19101 tasks each processing a block of 100 Mb input data. The median running time of one task was 1 hour 10 minutes. This corresponds to the processing time of about 1.4 Mb/min for such a median task and 0.84 Mb/min for on average for the entire corpus, including compression of the output CoNLL files (cf. Section 3.3.3.). The minimum time of processing of a task was 38 minutes while the maximum time was 9 hours and 4 minutes.

[20] `https://nlp.stanford.edu/software/stanford-dependencies.shtml`

[21] `http://jobimtext.org/dependency-collapsing`

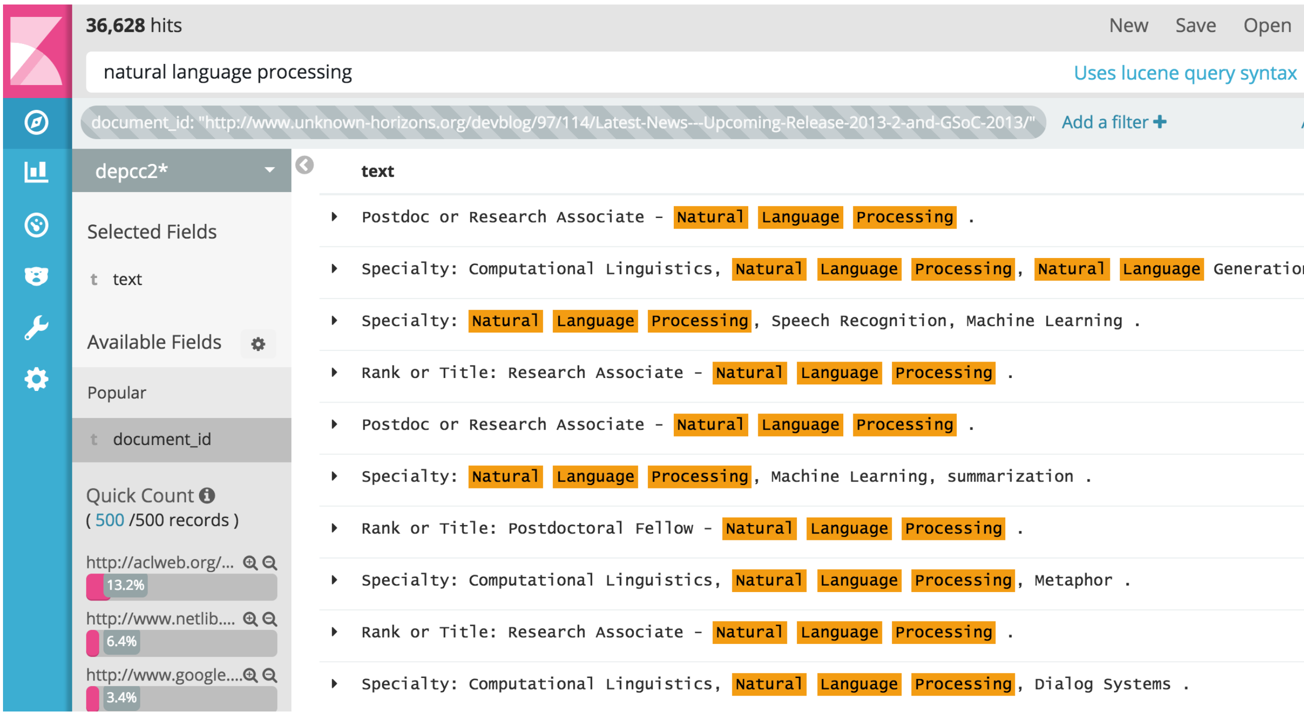

Figure 2: Interactive graphical user interface providing a full text search over 14.3 billion of sentences in the DEPCC corpus and their linguistic meta-data. A user can search for sentences containing specified keywords, named entities or syntactic dependencies.

3.7. Using the Corpus in the Amazon Cloud

The COMMON CRAWL datasets are hosted in the Amazon computing cloud platform.[22] As mentioned in the introduction, our corpus is also was made directly available on Amazon S3 distributed file system in the `us-east-1` region (US East, North Virginia) as a part of the COMMON CRAWL contributed datasets. This means that *you do not need to download the corpus* to be able to work with it. Instead, you can run the jobs directly against the respective bucket on the Amazon S3 file system which contains the DEPCC corpus. For optimal performance, you need to run instances which perform computations with the corpus inside the Amazon cloud (e.g. using the EC2[23] or EMR[24] services) in the `us-east-1` region.

3.8. Index of the Corpus

An access to a full-text search of all 14.3 billion sentences and their dependency relations of the DEPCC corpus is available upon request. This service is free and aims at facilitating access to the corpus for use-cases that do not require download of the entire collection of documents. Using the index, users can quickly retrieve sentences matching various linguistic criteria (using the Lucene query syntax[25]), e.g. presence of keyword in a sentence, presence of a specific named entity in a sentence, presence of a specific dependency relation, provenance of a document from a specific web domain, etc. Each retrieved sentence contains all the meta-data depicted in Table 3, such as the provenance of the sentence.

The corpus can be queried via a RESTful API based on the ElasticSearch search engine[26] or via a web-based graphical user interface based on Kibana[27] graphical interface to ElasticSearch. Results of a sample query visualized using Kibana are presented in Figure 2.

We do not distribute the index itself due to its huge size. However, users can re-create the index using the open source software provided as a part of the JoBimText package[28] from the the CoNLL files. While these compressed CoNLL files occupy only around 2.6 Tb, the size of the full index is about 15 Tb or more, depending on the replication factor of ElasticSearch. For this reason, for most practical applications, re-creation of the index is faster and more straightforward than download of a pre-computed index and its subsequent deployment. Besides, some major versions of the ElasticSeach indices are not compatible between one another.

4. Terms of Use

The DEPCC corpus is based on a COMMON CRAWL dataset. We do not reserve any copyrights as the authors of this derivative resource, but while using the DEPCC corpus you need to make sure to respect the Terms of Use of the original COMMON CRAWL dataset it is based on.[29]

[22] `https://aws.amazon.com`
[23] `https://aws.amazon.com/ec2`
[24] `https://aws.amazon.com/emr`
[25] `https://www.elastic.co/guide/en/kibana/current/lucene-query.html`
[26] `https://www.elastic.co`
[27] `https://www.elastic.co/products/kibana`
[28] `https://github.com/uhh-lt/josimtext`
[29] `http://commoncrawl.org/terms-of-use`

Model	SimVerb3500	SimVerb3000	SimVerb500	SimLex222
Wikipedia+ukWaC+BNC: Count SVD 500-dim (Baroni et al., 2014)	0.196	0.186	0.259	0.200
PolyglotWikipedia: SGNS BOW 300-dim (Gerz et al., 2016)	0.274	0.333	0.265	0.328
8B: SGNS BOW 500-dim (Gerz et al., 2016)	0.348	0.350	0.378	0.307
8B: SGNS DEPS 500-dim (Gerz et al., 2016)	**0.356**	**0.351**	0.389	0.385
PolyglotWikipedia:SGNS DEPS 300-dim (Gerz et al., 2016)	0.313	0.304	**0.401**	**0.390**
Wikipedia: LMI DEPS wpf-1000 fpw-2000	0.283	0.284	0.271	0.268
Wikipedia+ukWac+GigaWord: LMI DEPS wpf-1000 fpw-2000	0.376	0.368	0.419	0.183
DEPCC: LMI DEPS wpf-1000 fpw-1000	0.400	0.387	**0.477**	0.285
DEPCC: LMI DEPS wpf-1000 fpw-2000	**0.404**	**0.392**	**0.477**	**0.292**
DEPCC: LMI DEPS wpf-2000 fpw-2000	0.399	0.388	0.459	0.268
DEPCC: LMI DEPS wpf-5000 fpw-5000	0.382	0.372	0.442	0.226

Table 4: Evaluation results on the verb semantic similarity task. Sparse count-based distributional models (LMI) trained on the DEPCC corpus are compared to models trained on the smaller corpora, such as Wikipedia and a combination of Wikipedia, UKWAC, and GIGAWORD. Rows and columns of each LMI-weighted distributional model are pruned as in (Biemann and Riedl, 2013): the *wpf* indicates the number of words per feature, and the *fpw* indicates the number of features per word. We also compare our models to the best verb similarity models from the state of the art. Here the "BOW" denotes models based on bag-of-word features, while "DEPS" denotes syntax-based models. SimVerb3000 and SimVerb500 are train and test partitions of the SimVerb3500, while the SimLex222 dataset is composed of verb pairs from the SimLex999 dataset. The best results in a section are boldfaced, the best results overall are underlined.

5. Evaluation: Verb Similarity Task

As an example of potential use-case, we demonstrate the utility of the corpus and the overall methodology on a verb similarity task.

This task structurally is the same as the word similarity tasks based on such datasets as SimLex-999 (Hill et al., 2015). Namely, a system is given two words as input and needs to predict a scalar value which characterizes semantic similarity of the input words. While in the word similarity task the input pairs are words of various parts of speeches (nouns, adjectives, etc.), in this paper we only consider verb pairs.

We chose this task since verb meaning is largely defined by the meaning of its arguments (Fillmore, 1982), therefore dependency-based features seem relevant for building distributional representations of verbs.

5.1. Datasets: SimVerb3500 and SimLex222

Recently a new challenging dataset for verb relatedness was introduced, called SimVerb3500 (Gerz et al., 2016). The dataset is composed of 3500 pairs of verbs and is split into the train and test parts, called respectively SimVerb3000 and SimVerb500. In addition to this benchmark, in our experiments, we also test the performance of the models on the SimLex222, which is the verb part of SimLex999 dataset (Hill et al., 2015) composed of 222 verb pairs. Historically, the SimVerb3500 dataset was created after the SimLex222, addressing its shortcomings related to the verb coverage. As in our experiments, we do not use the dataset SimVerb3000 for training, and to be consistent with the results reported in (Gerz et al., 2016) we report performance of the tested verb similarity models on all four datasets: SimVerb3500/3000/500, and SimLex222.

5.2. A Distributional Model for Verb Similarity

We compute syntactic count-based distributional representations of words using the JoBimText framework (Biemann and Riedl, 2013).[30] The sparse vectors are weighted using the LMI weighting schema and converted to unit length. In our experiments, we varied also the maximum number of salient features per word (*fpw*) and words per feature (*wpf*). Conceptually, each row and column of the sparse term-feature matrix is pruned such that at most *wpf* nonzero elements in a row and *fpw* elements in a column are retained.

5.3. Discussion of Results

Table 4 presents results of the experiments.

5.3.1. Baselines

The top part of the table lists five top systems in various categories (Gerz et al., 2016), representing the current state-of-art result on this dataset. Namely, the Count based SVD system is from (Baroni et al., 2014). In the original paper, two corpora were used: the "8B" is a 8 billion tokens corpus produced by a script in the word2vec toolkit, which gathers the texts from various sources (Mikolov et al., 2013) and the "PolyglotWikipedia" is the English Polyglot Wikipedia corpus consisting of 1.9 billion tokens (Al-Rfou et al., 2013).

We use the baselines in the top of the table to indicate the best results on the dataset: our goal is to show the impact of the large corpora on performance and not to present a new model for verb similarity.

5.3.2. Impact of the Corpora on Performance

The bottom part of the table presents the distributional model described in Section 5.2. trained on the corpora of various sizes. Note, that the preprocessing steps for each corpus are exactly the same as for the DEPCC corpus. We observe that the smallest corpus (Wikipedia) yields the worst results. While the scores go up on the larger corpus, which is a combination of Wikipedia with two

[30] https://github.com/uhh-lt/josimtext

other corpora, we can reach the even better result by training the model (with exactly the same parameters) on the dependency-based features extracted from the full DEPCC corpus. This model substantially outperforms also the prior state of the art models, e.g. (Baroni et al., 2014) and (Gerz et al., 2016), on the SimVerb dataset, through the sheer size of the input corpus, as previously shown, e.g. (Banko and Brill, 2001) inter alia.

5.3.3. Differences in Performance for Test/Train Sets

For the SimVerb dataset, the absolute performance on the test part (SimVerb500) is higher than the absolute performance on the train part (SimVerb300) for almost all models, including the baselines. We attribute this to a specific split of the data in the dataset: our models do not use the training data to learn verb representations.

6. Conclusion

In this paper, we introduced a new web-scale corpus of English texts extracted from the COMMON CRAWL, the largest openly available linguistically analyzed corpus to date, according to the best of our knowledge.

The documents were de-duplicated and linguistically processed with part-of-speech and named entity taggers and a dependency parser, making it possible to easily start large-scale experiments with syntax-aware models without the need of long and resource-intensive preprocessing. We built an index of sentences and their linguistic meta-data accessible though an interactive web-based search interface or via a RESTful API.

In our experiments on the verb similarity task, a distributional model trained on the new corpus outperformed models trained on the smaller corpora, like Wikipedia, reaching new state of the art of verb similarity on the SimVerb3500 dataset. The corpus can be used in various contexts, ranging from training of syntax-based word embeddings (Levy and Goldberg, 2014) to unsupervised induction of word senses (Biemann et al., 2018) and frame structures (Kawahara et al., 2014). A promising direction of future work is using the proposed technology for building corpora in multiple languages.

7. Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) under the project "Joining Ontologies and Semantics Induced from Text" (JOIN-T). We are grateful to Amazon for providing required computational resources though the "AWS Cloud Credits for Research" program. Finally, we thank Kiril Gashteovski and three anonymous reviewers for their most helpful feedback.

8. Bibliographical References

Ballesteros, M. and Bohnet, B. (2014). Automatic feature selection for agenda-based dependency parsing. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING'2014)*, pages 794–805, Dublin, Ireland.

Banko, M. and Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France. Association for Computational Linguistics.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, volume 7, pages 2670–2676, Hyderabad, India. AAAI Press.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, MD, USA. Association for Computational Linguistics.

Biemann, C. and Riedl, M. (2013). Text: now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Biemann, C., Faralli, S., Panchenko, A., and Ponzetto, S. P. (2018). A framework for enriching lexical semantic resources with distributional semantics. *Natural Language Engineering*, pages 1–48.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Dohar, Qatar. Association for Computational Linguistics.

De Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of Conference on Language Resources and Evaluation (LREC'2006)*, pages 449–454. Genova, Italy, ELRA.

Eckart de Castilho, R. and Gurevych, I. (2014). A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Fillmore, C. J., (1982). *Frame semantics*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Hall, J., Nilsson, J., and Nivre, J. (2010). Single malt or blended? a study in multilingual parser optimization. In *Trends in Parsing Technology*, pages 19–33. Springer.

Kawahara, D., Peterson, D. W., and Palmer, M. (2014). A step-wise usage-based method for inducing polysemy-aware verb classes. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1030–1040, Baltimore, MD, USA. Association for Computational Linguistics.

Kolias, V., Anagnostopoulos, I., and Kayafas, E. (2014). Exploratory analysis of a terabyte scale web corpus. *arXiv preprint arXiv:1409.5443*.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, MD, USA. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., Harrahs and Harveys, NV, USA.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ruppert, E., Klesy, J., Riedl, M., and Biemann, C. (2015). Rule-based Dependency Parse Collapsing and Propagation for German and English. In *Proceedings of GSCL*, pages 58–66, Essen, Germany.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Tsuboi, Y. (2014). Neural networks leverage corpus-wide information for part-of-speech tagging. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 938–950, Doha, Qatar. Association for Computational Linguistics.

9. Language Resource References

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Gerz, D., Vulić, I., Hill, F., Reichart, R., and Korhonen, A. (2016). Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, TX, USA. Association for Computational Linguistics.

Goldberg, Y. and Orwant, J. (2013). A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Proceedings of *SEM in conjunction with NAACL-HLT*, pages 241–247, Portland, OR, USA. Association for Computational Linguistics.

Habernal, I., Zayed, O., and Gurevych, I. (2016). C4Corpus: Multilingual web-size corpus with free license. In *Proceedings of the Language Resources and Evaluation Conference (LREC'2016)*, pages 914–922, Portorož, Slovenia. ELRA.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with genuine similarity estimation. *Computational Linguistics*, 41(4):665–695.

Laippala, V. and Ginter, F. (2014). Syntactic N-gram collection from a large-scale corpus of internet Finnish. In *Human Language Technologies-The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT 2014*, volume 268, page 184, Kaunas, Lithuania. IOS Press.

Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. (2011). English Gigaword. *Linguistic Data Consortium*.

Schäfer, R. (2015). Processing and querying large web corpora with the COW14 architecture. In Piotr Baiaski, et al., editors, *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, pages 28–34, Lancaster. UCREL, IDS.