CoCo-Ex: A Tool for Linking Concepts from Texts to ConceptNet

Maria Becker, Katharina Korfhage, and Anette Frank
Department of Computational Linguistics, Heidelberg University, Germany
(mbecker, korfhage, frank)@cl.uni-heidelberg.de

Abstract

In this paper we present CoCo-Ex, a tool for **Ex**tracting **Co**ncepts from texts and linking them to the **Co**nceptNet knowledge graph. CoCo-Ex extracts meaningful concepts from natural language texts and maps them to conjunct concept nodes in ConceptNet, utilizing the maximum of relational information stored in the ConceptNet knowledge graph. CoCo-Ex takes into account the challenging characteristics of ConceptNet, namely that – unlike conventional knowledge graphs – nodes are represented as non-canonicalized, free-form text. This means that i) concepts are not normalized; ii) they often consist of several different, nested phrase types; and iii) many of them are uninformative, over-specific, or misspelled. A commonly used shortcut to circumvent these problems is to apply string matching. We compare CoCo-Ex to this method and show that CoCo-Ex enables the extraction of meaningful, important rather than overspecific or uninformative concepts, and allows to assess more relational information stored in the knowledge graph. [1]

1 Introduction

ConceptNet (Speer et al., 2017) is a semantic network which contains general commonsense facts about the world, e.g. *Birds can fly* or *Computers are used for sending e-mails* (Liebermann, 2008). It originates from the crowdsourcing project *Open Mind Common Sense* (Speer et al., 2008) that acquired commonsense knowledge from contributions over the web. The current version also includes expert-created resources such as WordNet (Fellbaum, 1998) and JMDict (Breen, 2004), other crowdsourced resources such as Wiktionary, knowledge obtained through games with a purpose such as Verbosity, and automatically extracted knowledge (cf. Speer et al. (2008)). Knowledge facts in ConceptNet are represented as triples, e.g. [*dog*,IsA,*domestic animal*]. The current version, ConceptNet 5, comprises 37 relations, such as UsedFor, IsA, PartOf, or LocatedAt.

ConceptNet has been proven a useful resource of background knowledge for various NLP downstream tasks, and is thus widely used, e.g., for reading comprehension (Mihaylov and Frank, 2018), machine comprehension (Wang et al., 2018; González et al., 2018), dialog modelling (Young et al., 2018), argument classification (Paul et al., 2020), textual entailment (Weissenborn et al., 2018), question answering (Ostermann et al., 2018) or for explaining sentiment (Paul and Frank, 2019).

As opposed to conventional knowledge bases such as NELL (Carlson et al., 2010), Freebase (Bollacker et al., 2008), or YAGO (Nickel et al., 2012), the nodes in ConceptNet are represented as non-canonicalized, free-form text. This means that (I) concept nodes are not normalized: e.g. *bake cake, bake cakes, baking cake*, and *baking cakes* are represented as distinct nodes; likewise *bin bag, binbag, bin bags*, and *bin-bag* are separate nodes in ConceptNet. (II) concept nodes often consist of multi-word expressions, which can be very long and complex. Often they consist of several nested phrase types, e.g., *buying the ingredients of the recipe*, or *a friend was celebrating a birthday*. (III) Since large parts of ConceptNet have been crowdsourced, it contains noise (e.g., typos), uninformative concepts (e.g., *there, it's*), or very specific concepts (e.g., the second concept in the triple: [*compute*,HasProp,*more complex than pencil*]).

These specific properties lead to a larger amount of nodes and a substantially sparser graph compared to conventional knowledge bases. This in turn is challenging for tasks such as knowledge

[1] We provide a demo video (https://www.youtube.com/watch?v=bgqVhE2vR9A&feature=youtu.be) and the code (https://github.com/Heidelberg-NLP/CoCo-Ex) for CoCo-Ex.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 119–126
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

base completion (cf. Li et al. (2016); Saito et al. (2018); Bosselut et al. (2019); Malaviya et al. (2020)); the semantic representation of nodes and edges (Speer and Lowry-Duda, 2017); or the learning of new relations (dos Santos et al., 2015; Becker et al., 2019; Trisedya et al., 2019).

Moreover, non-canonicalized nodes become challenging when merging knowledge bases, as in Faralli et al. (2020), who introduce a graph database merging multiple hypernymy graphs extracted from ConceptNet, DBpedia, WebIsAGraph, WordNet, and Wikipedia. They find that only 25% of the edges connect nodes from ConceptNet to other databases, which can be traced back to the fact that ConceptNet nodes are non-canonicalized, as opposed to common knowledge bases.

Finally, free-form concept nodes become problematic when we aim to project a ConceptNet subgraph from natural language texts by mapping phrases from natural language text to nodes in ConceptNet. In recent approaches, simple string matching has been applied to perform such a mapping (e.g. Lin et al. (2019); Wang et al. (2020)). Given the non-normalized nature of the concepts in ConceptNet, this can, however, result in an incomplete and noisy mapping: e.g., if the word “brains” occurs in a text, it can be mapped to the ConceptNet node *brains* (which is connected by 131 edges within ConceptNet), but not to *brain* (which is connected by 1799 edges). Therefore, a lot of relational knowledge stored in ConceptNet gets lost when mapping natural language text to concepts in ConceptNet via string matching. Moreover, since ConceptNet contains many nodes that don’t represent meaningful concepts (e.g. *yes, there, it’s, the*), simple string matching can lead to the extraction of concepts that will most likely be useless for downstream applications.

Motivated by these observations, we built a Concept Extraction Tool for ConceptNet, **CoCo-Ex**, which we present in this paper. CoCo-Ex is a tool written in Python 3.6 that selects *meaningful* concepts, possibly consisting of multiple tokens from natural language texts; it maps them to a *collection* of concept nodes in ConceptNet, utilizing the maximum of relational information stored in the knowledge graph. It is thus perfectly suited for identifying and extracting concepts from natural language texts and mapping them to ConceptNet, e.g., to project knowledge subgraphs from texts (Paul and Frank, 2019), or for detecting and classifying knowledge relations instantiated within texts (Becker et al., 2019).

We describe our Concept Extraction Tool CoCo-Ex in Section 2. In Section 3 we evaluate the benefits of CoCo-Ex in a practical application scenario, comparing it to simple string matching, by evaluating the retrieved concepts and their connectivity both automatically and manually. We conclude with a summary and results in Section 4.

2 CoCo-Ex: Extracting Concepts from Text and Mapping them to ConceptNet

CoCo-Ex is a pipeline implementation comprising several stages as shown in Figure 1.

In **Step 1**, we extract candidate phrases from a given text, which we preprocess in **Step 2**. In **Step 3**, we map the preprocessed phrases to ConceptNet concepts, which we preprocess in the same manner: We first create a dictionary based on ConceptNet, where we gather all concepts that are conceptually related (that is, referring to a similar or the same entity or event), but represented as distinct nodes. In this dictionary we then look up the preprocessed candidate phrases and get all ConceptNet nodes which contain them. In order to avoid obtaining conceptually unrelated nodes, in **Step 4** we establish a method that allows us to filter out nodes that are not similar enough to the candidate phrase using similarity metrics and vector space representations.

Step 1: Extracting Candidate Phrase Types. We start by extracting candidate phrases from a given text using the Stanford Constituency parser (Mi and Huang, 2015). We extract noun phrases, verb phrases and adjective phrases.[2] We find that some verb phrases are very long and specific and therefore unlikely to find exact matches in ConceptNet (e.g., “be sorted into different wheelie bins”). Yet, ConceptNet concepts often consist of general verb-object phrases, such as *walk the dog; cook dinner; bake a cake*. To accommodate for this, we create, for every verbal phrase we extract from the text, additional versions (i.e., chunks) that exclude subordinated prepositional phrases and/or noun phrases (e.g., for “be sorted into different wheelie bins” we additionally extract “be sorted into” and “be sorted”). Addressing the fact that nodes in ConceptNet are of different lengths and often consist

[2] We extract leaves (= tokens) of all subtrees that have one of the following phrase types or POS-tags: ’NP’, ’VP’, ’ADJP’, ’JJ’, ’JJR’, ’JJS’, ’NN’, ’NNS’, ’NNP’, ’NNPS’, ’VB’, ’VBG’, ’VBD’, ’VBN’, ’VBP’, ’VBZ’.

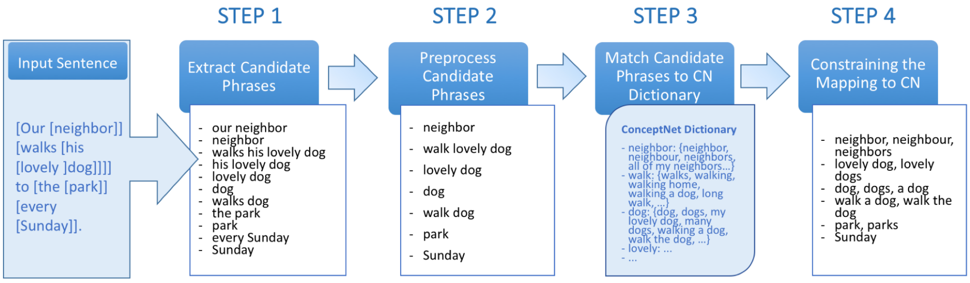
STEP 1
STEP 2
STEP 3
STEP 4
Input Sentence
[Our [neighbor]] [walks [his [lovely]dog]]]] to [the [park]] [every [Sunday]].
Extract Candidate Phrases
- our neighbor
- neighbor
- walks his lovely dog
- his lovely dog
- lovely dog
- dog
- walks dog
- the park
- park
- every Sunday
- Sunday
Preprocess Candidate Phrases
- neighbor
- walk lovely dog
- lovely dog
- dog
- walk dog
- park
- Sunday
Match Candidate Phrases to CN Dictionary
ConceptNet Dictionary
- neighbor: {neighbor, neighbour, neighbors, all of my neighbors...}
- walk: {walks, walking, walking home, walking a dog, long walk, ...}
- dog: {dog, dogs, my lovely dog, many dogs, walking a dog, walk the dog, ...}
- lovely: ...
- ...
Constraining the Mapping to CN
- neighbor, neighbour, neighbors
- lovely dog, lovely dogs
- dog, dogs, a dog
- walk a dog, walk the dog
- park, parks
- Sunday

Figure 1: Our pipeline for extracting and mapping phrases from texts to nodes in ConceptNet.

of several nested phrases, we keep all the original complex verbal phrases; the reduced chunks; and the split-off nested, subordinated phrases, which we again split into chunks (here: “different wheelie bins”, “wheelie bins”, and ”bins”).

Step 2: Preprocessing Candidate Phrase Types and ConceptNet Nodes. Next, we preprocess the candidate phrases we extracted from the text to prepare the mapping in Step 3. We apply spacy (Honnibal and Montani, 2017) to lemmatize the candidate phrases extracted from the texts, and remove articles, pronouns, adverbs, conjunctions, interjections and punctuation. The very same process we apply in Step 3 to nodes in ConceptNet, which are not normalized, in order to build a dictionary from ConceptNet.

Step 3: Matching Candidate Phrase Types to a Dictionary Based on ConceptNet. We then map the preprocessed phrases to the preprocessed ConceptNet concepts as follows: We create a dictionary based on ConceptNet where we collect all concepts that are conceptually related – in the sense that they involve at least one common content word – but are represented as distinct nodes in ConceptNet. I.e., we aim to subsume, e.g., *dog, dogs, nice dog,* and *my neighbour’s dog* under one entry in the dictionary (cf. Figure 2). In our dictionary, keys are lemmatized words contained in concept node phrases (e.g. ⟨dog⟩ for the concept *my dog*), and the corresponding value assigned to a key is a list of all ConcepNet nodes that contain this lemma (e.g. *dog, dogs, my dog, my neigbor’s dog*), as determined by the lemmatization of the nodes (see Step 2 for the applied process). Therefore, in our dictionary all ConceptNet nodes that contain the same lemma, the lemma of the key, are clustered together in one entry. Note that we lemmatize the ConcepNet nodes only for the purpose of mapping and clustering, while they remain unchanged (in their original form and inflection) as values in the dictionary. I.e., we compare a key (lemma) to the lemmatized version of the concepts, and include all nodes, or concept phrases in their original, inflected form, that contain this lemma.

An example of how we create an entry in the dictionary is given in Figure 2 and Figure 3: for the key ⟨dog⟩, all conceptually related nodes are retrieved from ConceptNet (Figure 2) by matching the (lemmatized) key and the lemmatized ConceptNet concepts (Figure 3, left side). All the retrieved ConceptNet nodes that contain the key lemma in their lemmatized form are stored as the key’s values (middle of Figure 3). In case the lemmatized candidate phrase from the text contains further lemmas, we apply the same procedure for each of these, and construct additional entries, if they have not yet been created and stored.

Using this dictionary we are now able to assess the maximum of relational information stored in the ConceptNet knowledge graph for a given candidate phrase from a text, since it allows us to jointly look up the in- and outgoing edges of all values (nodes) assigned to the same key, e.g., [*dogs*,ISA,*domestic animal*]; [*dog*,HASPROPERTY,*nice*]; ..) (Figure 3, right-hand side). In case a candidate phrase contains multiple lemmas, we collect the union of ConceptNet nodes defined for the respective lemmas (keys) as their values, and apply a filtering step, which we describe below, to select the concept nodes that best correspond to the complex phrase.

Specifically, when looking up extracted candidate phrases that contain *a single lemma* (e.g. ⟨dog⟩), we consider the complete list of nodes stored in the dictionary for that lemma (key) – that is, all concepts containing (inflected versions of) ⟨dog⟩, including also multiword phrases which are linked with other keys. When looking up extracted candidate phrases that contain *more than one lemma* (e.g. “walk the dog”), we obtain sets of ConceptNet nodes (values) that are defined for each (non-stopword) lemma (key) – here: ⟨dog⟩

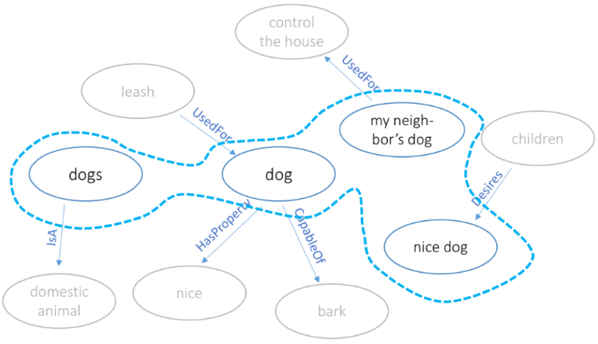

Figure 2: Collecting conceptually related nodes in ConceptNet, here: for the phrase ”the dog”.

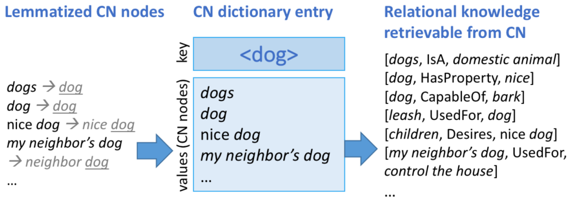

Figure 3: Example of the ConceptNet Dictionary entry for ⟨dog⟩. Left: lemmatized ConceptNet nodes (grey) that contain ⟨dog⟩ (underlined); middle: CN dictionary entry (containing the original CN nodes); right: relational knowledge (in- and outgoing edges for each value (CN node) assigned to the key) which can be retrieved from ConceptNet based on the dictionary entry.

and ⟨walk⟩ – and retrieve all ConceptNet nodes from their respective list of values. From these sets, instead of building their union, we construct their intersection, which yields the set of phrases from all keys' values that contain the maximum of lemmas contained in the candidate phrase.

For our example “walk the dog”, we would obtain the two lemmas ⟨walk⟩ and ⟨dog⟩, together with their values:

⟨walk⟩ → *walk, walks, walking, walking home,* ***walking a dog****, long walk,* ***walk the dog****, ...* ; and

⟨dog⟩ → *dog, dogs, nice dog, my neigbor's dog,* ***walking a dog****,* ***walk the dog****, ...*;

and extract *walking a dog* and *walk the dog* that are contained as values in both keys.

During the mapping process that collects values (ConceptNet concepts) for the lemmatized keys of candidate phrases, we are also resolving ambiguities. E.g., the forms *fly* or *flies* can be either a noun or a verb. We resolve this ambiguity by comparing the POS tags obtained during preprocessing the extracted candidate phrases to the POS tags that are associated with concepts in ConceptNet.[3] Specifically, we retrieve POS information for the extracted candidate phrases by applying the POS tagger implemented in spacy (Honnibal and Montani, 2017) on the sentence level, while for ConceptNet nodes we assess the POS labels available as metadata. In case we find several concepts with the same surface form but different POS tags in ConceptNet (e.g. *fly/noun* and *fly/verb*), we use the POS annotations from the extracted candidate phrases and from ConceptNet tags to restrict the mapping to matching POS, hence we do not include any concepts with conflicting POS information in the list of values for the phrase's keys.

To summarize, the dictionary we obtain from Step 3 allows us to look up concepts for any preprocessed candidate phrases, and obtain from it all ConceptNet nodes which contain them or inflected versions of them. In case of multiple lemmas contained in a candidate phrase, we retrieve all nodes that contain *all lemmas* included in the given phrase, by computing an intersection over the values associated with all keys (lemmas) evoked by the phrase.[4] Since we lemmatize both the ConceptNet nodes and the extracted candidate phrases as described above, we maximize the number of matches, and hence, the associated ConceptNet relation tuples, while selecting maximally specific nodes. At the same time, since we construct *chunked* phrases from the extracted concepts, we also allow for more constrained matches (limited, e.g., to single lemmata) with equally constrained ConceptNet concepts, preventing over-specific phrases and an ensuing loss of recall. Finally, we apply POS filtering, and hence avoid the retrieval of ConceptNet concepts that do not match the POS category of the concepts mentioned in the candidate phrase, relying on the sentential context of the phrase candidate for disambiguation.

Step 4: Constraining the Mapping to ConceptNet Concepts. While in Step 3 we constrain the selected concept nodes by intersection in case the phrase candidate contains multiple lemmata, we still obtain many ConceptNet nodes when mapping short phrases containing a single content word to ConceptNet, since we retrieve all nodes that include the lemma of the candidate phrase. In practice, this yields a huge set of concepts that contain

[3] We find POS information for a majority of concepts contained in ConceptNet, as used in specific tuples. In cases where this information is not given, we do not apply any filtering.

[4] This holds as long as the lemmas identified in the textual phrases can be identified within ConceptNet's concept nodes.

not only this lemma, but many other content words not present in the candidate phrase – possibly conceptually unrelated nodes that we want to omit. For example, if the candidate phrase is “dog”, we map it to the ConceptNet nodes *dog* and *dogs*, but also conceptually not strictly related nodes such as *feeding my dogs*, *dogs are my favourite animals*, *it’s raining cats and dogs*, etc. We therefore establish a method that allows us to filter out nodes that are not similar enough to the candidate phrase, and hence are assumed to be conceptually unrelated, which we describe in the following.

We filter the nodes (values) for each lemma (key) by calculating the similarity between the ConceptNet concepts and the extracted candidate phrase. We calculate similarity in terms of length (by token or char length) and in terms of semantic similarity (using word embeddings and similarity metrics). We experimented with different similarity metrics: we tried Dice Coefficient (Sørensen, 1948), Jaccard Coefficient (Jaccard, 1902), Minimum Edit Distance, Word Mover’s Distance (Kusner et al., 2015), and Cosine Distance, with different similarity thresholds. For the metrics that require word representations in vector space (Word Mover’s Distance and and Cosine Distance), we tried different embeddings (Numberbatch (Speer et al., 2017), Word2Vec trained on GoogleNews (Mikolov et al., 2013), and GloVe (Pennington et al., 2014)), where we compute representations for multiword terms by averaging their embeddings. We also consider differences in phrase lengths: here we compare the length of the ConceptNet nodes’ concept phrases to the length of the candidate phrase – by number of tokens and of characters. E.g. when comparing the candidate phrase ”my dog” to the nodes (a) *dogs* and (b) *many dogs*, we obtain for (a) a difference in the number of tokens by 1 and of characters by 1, and for (b) in the number of tokens by 0 and of characters by 3.

We evaluated the output of several configurations manually in terms of how well the filtered nodes fit the extracted candidate phrase, and found the following configurations to yield the highest coverage and lowest noise: we allow for a maximum token length difference of 1 and/or a maximum character difference of 10, and a minimum Dice coefficient of 0.85. The other configurations are implemented as well (as command line parameters), so users can experiment with different settings easily.

		Str-Match	**CoCo-Ex**
CommonsenseQA	Questions	99,217	88,631
	Answers	106,681	116,941
OpenBookQA	Questions	38,415	38,485
	Answers	53,748	61,313

Table 1: Number of concepts linked to ConceptNet by simple string matching vs. using CoCo-Ex. CommonsenseQA contains 12,247 questions with 5 answer choices each, and OpenBookQA provides 6,000 4-way multiple-choice questions.

3 Applications

Recent approaches that map natural language text to nodes in ConceptNet apply simple string matching. Wang et al. (2020) for example use ConceptNet in order to retrieve multi-hop knowledge paths as background information for improving the task of question answering. They map concepts that appear in questions and answers from the two benchmark datasets, CommonsenseQA (Talmor et al., 2019) and OpenBookQA (Mihaylov et al., 2018), to ConceptNet using plain string matching.

Irrespective of the question answering task, we want to evaluate the two methods of linking concepts from texts to ConceptNet (plain string matching vs. CoCo-Ex) by comparing the **number of concepts** that could be retrieved from ConceptNet by both methods, respectively; and by evaluating the **quality** of the retrieved concepts, with regard to their coverage and informativity, as well as the amount of utilized relational knowledge from the ConceptNet knowledge graph.

We reimplement the string matching method and make it comparable to CoCo-Ex by retrieving all noun phrases, verb phrases and adjective phrases and their nested phrases (as we do for CoCo-Ex). Additionally, as in CoCo-Ex, we filter these phrases by removing articles, pronouns, adverbs, conjunctions, interjections and punctuation, and keep the original phrases and the chunked versions.

The **counts of concepts** retrieved by simple string matching vs. using CoCo-Ex are displayed in Table 1. We find that for the CommonsenseQA dataset, more concepts are linked to ConceptNet from the questions when using string matching, while with CoCo-Ex we can link more concepts from the answers (Table 1). For OpenBookQA, the number of extracted concepts for the questions are similar for both methods, while again we can link more concepts from the answers with CoCo-Ex.

For evaluating **concept quality**, we set up a

	Str-Matching	CoCo-Ex
CommonsenseQA		
Coverage (binary)	17 of 25 (68%)	20 of 25 (80%)
Ratio of Informative (*Wanted*) Concepts (total and %)	152 of 220 (69%)	190 of 192 (99%)
Connecting Edges of Informative Concepts (total/avg-question)	151,526/6,061	185,663/7,427
OpenBookQA		
Coverage (binary)	16 of 25 (64%)	14 of 25 (56%)
Ratio of Informative (*Wanted*) Concepts (total and %)	92 of 148 (62%)	104 of 145 (72%)
Connecting Edges of Informative Concepts (total/avg-question)	91,938/3,678	110,039/4,402

Table 2: Manual evaluation of linked concepts from 25 questions for each dataset. For each question, our annotators evaluated if all meaningful concepts were extracted (*Coverage*; in a binary evaluation setup *yes/no*); and how many of the extracted concepts are informative (*wanted*) (*Ratio wanted/wanted+unwanted*) . For all informative (*wanted*) concepts, we then looked up the number of edges connecting these nodes in ConceptNet (in- and outgoing *edges*).

small annotation experiment where we provided our annotators with 50 questions randomly sampled from CommonsenseQA and OpenBookQA. For each question, our annotators evaluated whether all meaningful concepts were extracted (coverage, in a binary setting (yes/no)); and if/how many informative (and thus, *wanted*) concepts are among the extracted concepts (which can be interpreted as reverse precision).[5] For each dataset, two annotators with linguistic background performed annotations. We measure annotator agreement in terms of Cohen's Kappa and achieve an agreement of 78%. Remaining conflicts were resolved by an expert annotator (one of the authors). The number of concepts that could be accessed in ConceptNet we evaluate automatically, in terms of the number of in- and outgoing edges connecting the node(s) which have been annotated as informative (*wanted*), identified by simple string matching vs. all nodes obtained by CoCo-Ex through keys and values.

The results of our manual evaluation experiment are displayed in Table 2. We find that the coverage (if all meaningful concepts were extracted, evaluated in a binary setting: yes/no) is higher for CommonsenseQA when using CoCo-Ex and higher for OpenBooksQA when applying string matching.

Next, we evaluate the informativeness of the extracted concepts. We find that the ratio between informative (*wanted*) and uninformative concepts (*unwanted*) is much better when using CoCo-Ex opposed to simple string matching on both datasets (cf. Table 2). Finally, we also evaluate the amount of relational information stored in the ConceptNet knowledge graph which can be retrieved by looking up in- and outgoing nodes from the nodes rated as informative. Here we find that with CoCo-Ex, much more relational information of ConceptNet can be accessed, indicating again the superiority of this method compared to simple string matching.

4 Conclusion

In this paper we presented CoCo-Ex, a tool for **Ex**tracting **Co**ncepts from texts and linking them to the **Co**nceptNet knowledge graph. As opposed to the common shortcut method of simply matching strings from texts to ConceptNet nodes, CoCo-Ex extracts *meaningful* concepts from texts and maps them to *collections of* concept nodes in ConceptNet, which enables us to assess the maximum of relational information stored in the ConceptNet knowledge graph. CoCo-Ex takes into account that concepts in ConceptNet are represented as non-canonicalized, free-form text and are often complex, noisy, uninformative, and/or over-specific. We evaluated CoCo-Ex against the method of simple string matching, which confirmed our hypotheses that (i) CoCo-Ex improves the precision of mapping by enabling the extraction of meaningful, important rather than overspecific or uninformative concepts, and (ii) allows to utilize the maximum of relational information stored in the knowledge graph, a step towards overcoming the well-known sparsity issue of commonsense knowledge graphs such as ConceptNet.

Acknowledgements

This work was funded by the DFG within the project ExpLAIN as part of the Priority Program *RATIO* (SPP-1999). We thank our annotators for their contribution.

[5] Our annotation manual can be found here: https://github.com/Heidelberg-NLP/CoCo-Ex/blob/master/CoCo-Ex_Annotation_Manual.pdf

References

Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. 2019. Assessing the difficulty of classifying ConceptNet relations in a multi-label classification setting. In *RELATIONS - Workshop on meaning relations between phrases and sentences*, pages 1–15, Gothenburg, Sweden. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Jim Breen. 2004. JMdict: a Japanese-multilingual dictionary. In *Proceedings of the Workshop on Multilingual Linguistic Resources*, pages 65–72, Geneva, Switzerland. COLING.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313. AAAI Press.

Stefano Faralli, Paola Velardi, and Farid Yusifli. 2020. Multiple knowledge GraphDB (MKGDB). In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2325–2331, Marseille, France. European Language Resources Association.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

José-Ángel González, Lluís-F. Hurtado, and Ferran Pla. 2018. ELiRF-UPV at SemEval-2019 task 3: Snapshot ensemble of hierarchical convolutional neural networks for contextual emotion detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 195–199, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Paul Jaccard. 1902. *Lois de distribution florale dans la zone alpine*, volume 38. Bulletin de la Société Vaudoise des Sciences Naturelles.

Matt J. Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 957–966.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense Knowledge Base Completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Henry Liebermann. 2008. Usable AI Requires Commonsense Knowledge. In *Workshop on Usable artificial intelligence, held in conjunction with the Conference on Human Factors in Computing Systems (CHI)*, pages 1–5.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Choi Yejin. 2020. Commonsense knowledge base completion with structural and semantic context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2925–2933.

Haitao Mi and Liang Huang. 2015. Shift-reduce constituency parsing with dynamic programming and POS tag lattice. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1030–1035, Denver, Colorado. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, pages 1–12.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: Scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 271–280, New York, NY, USA. Association for Computing Machinery.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, New Orleans, Louisiana. Association for Computational Linguistics.

Debjit Paul and Anette Frank. 2019. Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, volume 1, pages 3671–3681, Minneapolis, Minnesota, USA.

Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. Argumentative Relation Classification with Background Knowledge. In *Proceedings of the 8th International Conference on Computational Models of Argument (COMMA 2020)*, pages 319–330.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2018. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150, Brussels, Belgium. Association for Computational Linguistics.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of 31St AAAI Conference on Artificial Intelligence*, pages 444–451.

Robyn Speer, Catherine Havasi, and Henry Lieberman. 2008. AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, pages 548–553. AAAI Press.

Robyn Speer and Joanna Lowry-Duda. 2017. ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 85–89, Vancouver, Canada. Association for Computational Linguistics.

Thorvald Sørensen. 1948. *A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons*. Kongelige Danske Videnskabernes Selskab. 5 (4): 1–34.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy. Association for Computational Linguistics.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018. Yuanfudao at SemEval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 758–762, New Orleans, Louisiana. Association for Computational Linguistics.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

Dirk Weissenborn, Tomas Kocisky, and Chris Dyer. 2018. Dynamic Integration of Background Knowledge in Neural NLU Systems. In *International Conference on Learning Representations (ICLR) 2018*.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting End-to-End Dialog Systems with Commonsense Knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4970–4977.