Extraction of Common-Sense Relations from Procedural Task Instructions using BERT

Viktor Losing, Lydia Fischer, Jörg Deigmöller

2021

Preprint:

This is an accepted article published in International Global Wordnet Conference. The final authenticated version is available online at: https://doi.org/[DOI not available]

Extraction of Common-Sense Relations from Procedural Task Instructions using BERT

Viktor Losing, Lydia Fischer, Joerg Deigmoeller

Honda Research Institute Europe
Carl Legien Strasse 17, Offenbach, Germany

Abstract

Manipulation-relevant common-sense knowledge is crucial to support action-planning for complex tasks. In particular, instrumentality information of what can be done with certain tools can be used to limit the search space which is growing exponentially with the number of viable options. Typical sources for such knowledge, structured common-sense knowledge bases such as ConceptNet or WebChild, provide a limited amount of information which also varies drastically across different domains. Considering the recent success of pre-trained language models such as BERT, we investigate whether common-sense information can directly be extracted from semi-structured text with an acceptable annotation effort. Concretely, we compare the common-sense relations obtained from ConceptNet versus those extracted with BERT from large recipe databases. In this context, we propose a scoring function, based on the WordNet taxonomy to match specific terms to more general ones, enabling a rich evaluation against a set of ground-truth relations.

1 Introduction

Lately, AI-based methods advanced rapidly in their capabilities leading to the tackling of ever more challenging tasks. This progress is expected to continue and to potentially culminate in general-purpose intelligence within a few decades (Müller and Bostrom, 2016). However, current systems are designed for highly specific tasks and lack crucial capabilities for their application in a broader context. In particular, they have insufficient common-sense knowledge and reasoning capabilities. Common-sense knowledge are essential facts humans acquire throughout our life and which are frequently applied for everyday tasks, mostly in a subconscious manner. However, such knowledge is rarely expressed as it is unnecessary to state the obvious, making it highly elusive. This paper addresses this challenge and focuses on the acquisition of common-sense knowledge. Concretely, we are interested in manipulation-relevant knowledge that can support action planning, answering questions such as "What do I need to cut a bread?" or "What can I do with a knife?". The acquisition is framed as a relation-extraction task where we focus on the instrumentality relations between tools, actions and objects in the kitchen domain.

The classical approach to acquire common sense knowledge is to query publicly available knowledge bases. However, since common-sense information is scarcely expressed, there exist only a few dedicated sources (Speer and Havasi, 2013; Tandon et al., 2017). The most prominent one is the ConceptNet knowledge graph (Speer and Havasi, 2013), which is widely used in various applications (Camacho-Collados et al., 2017; Bosselut et al., 2019; Mihaylov and Frank, 2018). It connects words and phrases of natural language with labeled assertions, so-called predicates. The main issue of ConceptNet and similar sources is that the provided amount of information is rather limited, due to the fact that crowd-sourcing is an ineffective strategy for collecting common-sense information. The incentive for the public to contribute is weak as the information by definition is common sense and widely known. Furthermore, the amount and granularity of the information varies substantially across different topics, making it rather unclear to assess its relevance for specific applications.

Recently published language models such as BERT or GPT2 (Devlin et al., 2018; Radford et al., 2018) constitute a drastic leap forward in the domain of natural language processing (NLP) as they distinctly improved benchmarks in the tasks of

language translation, question answering, named entity recognition and many more. These large neural networks are pre-trained on a massive amount of unsupervised data and can be fine-tuned to different tasks based on comparably few labeled examples. Considering their success story, it is interesting to investigate their effectiveness in the extraction of common-sense information from widely available text databases. Such an approach is scalable as the models can easily process additional sources.

In this paper, we apply this concept to acquire instrumentality relations from procedural task instructions, more specifically recipes. Procedural task instructions are one of the few sources where common-sense knowledge is made explicit because they aim to instruct humans to perform a task they are potentially unfamiliar with. Concretely, we fine-tune BERT to learn the relation extraction using a few labeled examples and compare the yielded relation set against the one of ConceptNet. The evaluation is based on a set of ground-truth relations, which we collect in a study. In this context, we propose a scoring function to match specific terms to more general ones based on the WordNet taxonomy. The extensive evaluation underlines the effectiveness of BERT, leading to distinctly more relations with an acceptable proportion of false relations that can flexibly be adjusted with standard filtering techniques.

2 Related Work

The lack of common-sense knowledge and reasoning capabilities was recently addressed in DARPA's "Machine common sense" initiative (Gunning, 2018) and led to an increased attention within the research community. Various new sources for common-sense knowledge have been established since. ATOMIC (Sap et al., 2019), for example, provides a database with causes and effects of common everyday actions such as making a coffee. WebChild (Tandon et al., 2017) is a common-sense knowledge graph that provides in contrast to other sources also comparative knowledge using relations such as "larger than" between different concepts. It does not rely on crowd-sourcing, but instead uses different algorithms to accumulate the knowledge on the basis of large text corpora. Databases for visual common-sense have been recently proposed by Goyal et al. (2017).

Sources derived from Wikipedia such as DBPedia (Lehmann et al., 2015), Yago (Suchanek et al., 2007) or WikiData (Vrandecic and Krötzsch, 2014) have often been used to extract common-sense knowledge from. Jebbara et al. (2019) proposed multiple score-functions in to rank relations that encode prototypical locations of objects. They relied on crowd-sourcing, DBPedia and annotated image databases to generate ground truth relations to evaluate their methods. Manipulation-related knowledge is particularly interesting in the field of robotics, where explicit action representations are based on relations between one action and the manipulated object (Zech et al., 2019). Such relations are extracted from video (Yang et al., 2014), text data (Jebbara et al., 2019; Kaiser et al., 2014) or even multiple modalities (Yang et al., 2016).

Common-sense knowledge is tackled in a broad range of topics. Zhou et al. tackled temporal common-sense by proposing dedicated datasets and a specific language model that outperforms BERT on the task of classifying typical events according to their temporal properties (Zhou et al., 2019; Zhou et al., 2020). Common-sense properties of word embeddings were extracted by Yang et al. (2018) using a zero-shot learning approach. This enables a property-based comparison of entities to answer questions like "Is an elephant bigger than a tiger?". Hu et al. (2019) augmented the entities contained in the SQuAD dataset (Rajpurkar et al., 2016) with common-sense knowledge from ConceptNet and WordNet, allowing them to answer a variety of additional questions about the entities.

Recent work focuses on the extraction of action effects, i. e. how does the object state changes when certain actions are applied (Gao et al., 2018). In this regard, the action context is often encoded as well (Baker et al., 1998; Palmer et al., 2005; Yang et al., 2016; Chai, 2018), which is similar to the linguistic concept of verb semantics (Wu and Palmer, 1994). Verb semantics describe the meaning of a verb within a context depending on the "agent" (the one executing the action), the "patient" (here the object on which the action is applied on) and an instrument (the tool used for manipulation).

Fine-tuning BERT has been done for various tasks. Recently, Wang et al. (2019) proposed a two-step process for entity relation extraction from documents and argued for its adoption as new task baseline, since it clearly outperformed the current baseline approach (LSTM).

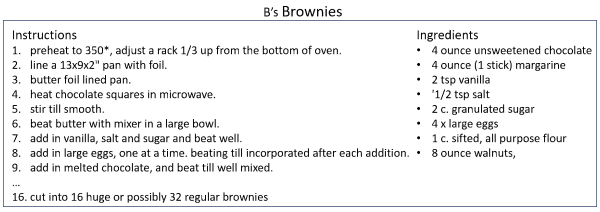
In contrast to the mentioned work, our contribution provides three novel aspects. First, it in-

vestigates the viability of common-sense relation extraction using pre-trained models that are fine-tuned with a very small amount of labeled examples for a specific application. Second, it measures the relevance of the extracted relations based on their coverage of a ground-truth relation set, thereby proposing a scoring function to consider the matching between specific and more general terms. Lastly, it compares the relation set against the one contained in ConceptNet, which provides insight into ConceptNet's practical relevance for the specific application.

3 Approach

We are interested in acquiring instrumentality relations for the kitchen domain. Specifically, we want to know the relevant tools for certain tasks. For instance, a knife can be used for cutting bread, but a cutting board may be helpful as well. A relation $r = (t, a, o)$ is a triplet consisting of three strings, where t encodes one tool and the associated task is described by action a and object o. Some examples for relevant relations are (knife, cut, bread), (fridge, cool, food), and (bowl, mix, salad).

We use BERT to extract such information from large text corpora, where the text is loosely structured. In the past, powerful models required a large amounts of labeled data to achieve a reasonable performance, making such approaches not applicable for most applications. However, pre-trained models have drastically reduced the label-burden and simultaneously increased their performance. The main advantage of this approach in comparison to the extraction from structured database is its scalability. Once the model is trained it can easily be applied on vast amounts of available text to harness the desired information. Hence, more relations can be extracted with a higher language variety as are contained in current common-sense databases. Furthermore, it can be applied on other domains as long as text corpora cover the relevant relations. The disadvantage is the necessity of annotating some examples for the specific application. In the following, we describe the approach in detail and also propose an evaluation metric to measure the match between two relation sets. This is crucial to determine the relevance of the extracted relations according to a set of ground-truth relations.


B's Brownies

Instructions
1. preheat to 350*, adjust a rack 1/3 up from the bottom of oven.
2. line a 13x9x2" pan with foil.
3. butter foil lined pan.
4. heat chocolate squares in microwave.
5. stir till smooth.
6. beat butter with mixer in a large bowl.
7. add in vanilla, salt and sugar and beat well.
8. add in large eggs, one at a time. beating till incorporated after each addition.
9. add in melted chocolate, and beat till well mixed.
...
16. cut into 16 huge or possibly 32 regular brownies

Ingredients
- 4 ounce unsweetened chocolate
- 4 ounce (1 stick) margarine
- 2 tsp vanilla
- '1/2 tsp salt
- 2 c. granulated sugar
- 4 x large eggs
- 1 c. sifted, all purpose flour
- 8 ounce walnuts,

Figure 1: One recipe of *Recipe1M+*. Only a few instructions contain a complete relation.

3.1 Relation Extraction from Recipes

An obvious source for task-specific relations are procedural task instructions from the task domain. The instructions decompose the description of how to complete a task in a step-wise manner. Single steps are phrased in a brief way, only specifying the necessary information. Examples for procedural task instructions are do-it-yourself manuals or recipes. These are nowadays publicly available for a broad range of tasks and domains. WikiHow (Koupaee and Wang, 2018) for instance is a webpage that provides procedural-task description for a broad range of everyday tasks such as "How to clean a kitchen table?", but also very specific ones as "How to take the U.S. census?".

As we are interested in the kitchen domain, we rely on the large recipe database *Recipe1M+* (Marin et al., 2019). It contains over 1 Million cooking recipes covering a broad range of topics and themes with a high variety of used language. Figure 1 shows a recipe example consisting of the ingredients and the instructions. We only use the latter. Even though procedural task instructions are usually densely packed with relations, the extraction is still a challenging problem as these are phrased in a peculiar language, often neglecting a valid English grammar. Instructions can be very brief, use domain-specific terms and often require the context of previous steps for resolving ambiguous references.

3.1.1 Token Classification

We frame the relation extraction from instructions as a token-classification task, where tools, actions, and objects are mapped to their respective token labels. A set of labeled instructions is used to fine tune BERT. From each instruction at most one relation is extracted. The model solely accesses the single instruction, i. e. it does not consider previous instructions. In fact, most instructions do not explicitly name a complete relation as can be seen in the example of Figure 1. Only instructions 4 and

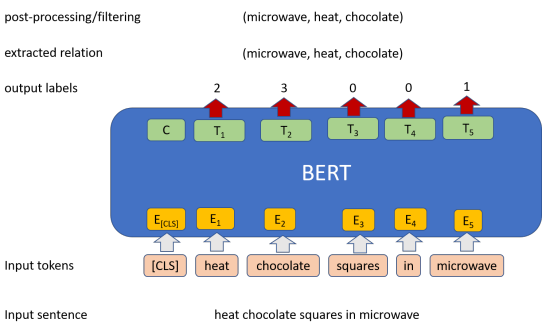


Figure 2: Overview of the processing pipeline.

6 contain a complete relation, whereas the others provide partial relations, requiring previous instructions or common-sense information to fill the gap. In such cases, all tokens are labeled as irrelevant. Conversely, there are instructions that contain more than one complete relation, if multiple or alternative tools are suggested to perform a task. For instance, instruction 6 of the exemplary recipe names a *mixer* and a *large bowl* as required tools to beat the butter. Here, we simply label the relation that gets mentioned first and ignore the other. This limitation of the token-classification was accepted in favour of its simplicity as our focus is to demonstrate easy-to-use alternatives to common-sense knowledge bases. Nonetheless, there is uncovered potential to increase the data efficiency of the models by applying more sophisticated architectures that are able to extract relations across instructions or consider multiple relations per instruction. Altogether, we annotated 400 instructions for fine-tuning of which 230 contain a valid relation.

Figure 2 shows the pipeline of the relation extraction. An instruction is tokenized and fed into the fine-tuned BERT model. The related output labels are concatenated to the relation structure and validated by a post-processing step.

3.2 Post-processing

Some of the extracted relations are filtered or modified to reduce the amount of false relations. Formally, let V be the set of all WordNet lemmas that are assigned to verb-synsets and N the ones assigned to noun-synsets. Furthermore, let T be the set of predefined tools. A given relation $r = (t, a, o)$ is only kept if:

$$a \in V \wedge o \in N \wedge \exists t_i \in T : substring(t_i, t),$$

where $substring(a, b)$ is a boolean function that determines whether a is a substring of b.

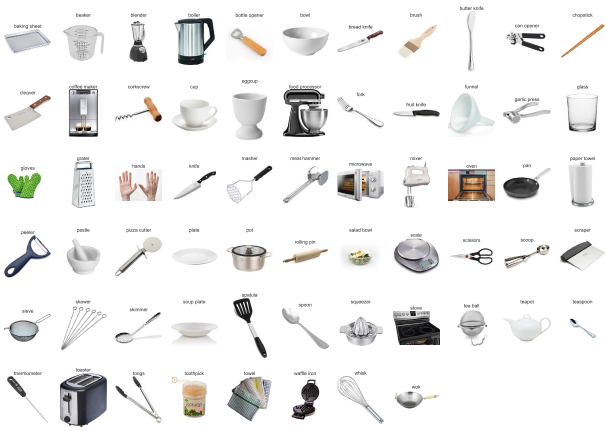


Figure 3: The set of predefined tools.

Assuming the relation is kept, we still have to map t to a concrete tool $\hat{t} \in T$. Here, our goal is to conserve the specificity of the extracted tool. For instance, assume t = "big fruit knife" than we prefer to map it to "fruit knife" over the more general term "knife". Therefore, we choose the $\hat{t} \in T$ that is the longest substring of t, i. e. $\hat{t} = \arg\max_{t_i \in T} \mathbb{1}(substring(t_i, t))|t_i|$.

3.3 Ground-Truth Relations

There are various possibilities to estimate the quality of common-sense relations. One viable approach is to estimate the number of "correct" relations, based on sampling and manual inspection. However, it neglects the quality aspect as some relations are clearly more common and intuitive than others. Instead, we count the amount of matched ground-truth relations, that were proposed in a study as common relations. We predefined 63 tools as depicted in Figure 3, and asked ten subjects to provide relations for those.

Neither the actions nor the corresponding objects were restricted, but we provided a few instructive examples. The subjects had 20 minutes to come up with as many relations as possible. Altogether, 539 relations were collected of which 386 are unique. Table 1 shows the most frequently named relations.

3.4 Relation Matching

Given a set of m ground-truth relations $G := \{(t_1, a_1, o_1), \ldots, (t_m, a_m, o_m)\}$ and a set of n candidate relations $C = \{(\hat{t}_1, \hat{a}_1, \hat{o}_1), \ldots, (\hat{t}_n, \hat{a}_n, \hat{o}_n)\}$ we want to measure the matching error $e(G, C)$. The naive approach is to use the intersection of both sets:

Table 1: The most common relations provided by the subjects.

Relation	Recurrence
can opener, open, can	9
masher, mash, potato	9
garlic press, press, garlic	8
bread knife, cut, bread	7
grater, grate, cheese	7
coffee maker, make, coffee	6
corkscrew, open, wine bottle	5
oven, bake, cake	5
pizza cutter, cut, pizza	5
bottle opener, open, bottle	4

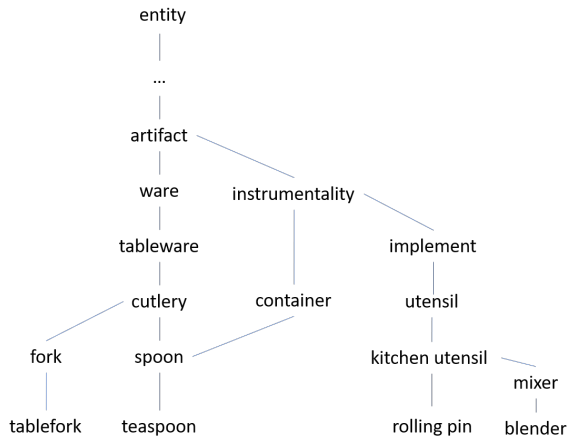


Figure 4: A fraction of the WordNet taxonomy covering a few of the predefined tools.

$e(G, C) = 1 - |G \cap C|/|G|$. However, such a measure accounts only for perfect matches, whereas it is reasonable to match against synonyms or semantically close terms. Hence, we propose a more informative matching function that utilizes the hypernym graph of WordNet. The intuition behind the matching is that a ground-truth relation can always be generalized, but never specialized. This is illustrated based on the depicted hypernym graph in Figure 4. The ground-truth relation (knife,cut,bread) can be generalized to (cutlery,cut,bread) or even (entity,cut,bread), as there is at least one hyponym of cutlery, knife itself, that is able to perform the task. In contrast, specialization entails that *all* instances can be used in such a context, which usually does not apply. In other words, if (cutlery,cut,bread) is the ground-truth relation we cannot conclude that all hyponyms of cutlery can be used as well, as spoon for instance is unsuitable. Consequently, the matching function between two relations cannot be symmetric, since it is crucial to distinguish between the ground-truth and the candidate. In other words, it is a pseudo-metric. Nonetheless, as the distance notion is quite intuitive we keep it throughout the paper. In the following, we first define the matching for single words and subsequently for whole relations.

3.4.1 Word Distance

Let each word w be assigned to a set of synsets S_w, where $S_w = \emptyset$ for words that are not represented in WordNet. The distance between the ground-truth word w and the candidate word $\hat{w}$ is the minimum distance between their synsets S_w, $S_{\hat{w}}$:

$$d(w, \hat{w}) = \min_{\forall s \in S_w, \forall \hat{s} \in S_{\hat{w}}} \hat{d}(s, \hat{s}) \tag{1}$$

Every synset s has a set of hypernym paths $P_s = \{p_1, \ldots, p_k\}$, where each path $p_i := [s_1, s_2, \ldots, s_k | s_1 = s \wedge s_k = r]$ connects s with the root synset r ("entity" is the root synset in WordNet). We denote the distance between a synset s and a path p as the index of s in p:

$$\tilde{d}(s, p) = \begin{cases} Index(s, p) & \text{if } s \in p \\ \infty & \text{otherwise.} \end{cases}$$

Finally, we are able to define the distance between a ground-truth synset s and a candidate synset $\hat{s}$

$$\hat{d}(s, \hat{s}) = \min_{\forall p \in P_s} \tilde{d}(\hat{s}, p).$$

3.4.2 Relation Distance

The distance between a ground-truth relation $r = (t, a, o)$ and the candidate $\hat{r} = (\hat{t}, \hat{a}, \hat{o})$ is simply the element-wise sum of word distances

$$D(r, \hat{r}) = d(t, \hat{t}) + d(a, \hat{a}) + d(o, \hat{o}).$$

The distance measure can be used to parameterize the error rate function with a maximum relation distance k:

$$e_k(S, \hat{S}) = \frac{1}{|S|} \sum_{i=1}^{m} \mathbb{1}(\arg\min_{j \in \{1, \ldots n\}} D(S_i, \hat{S}_j) > k) \tag{2}$$

Varying k allows to control the matching granularity, i. e. $k = 0$ considers only perfect matches or those using the WordNet synonyms, whereas $k \to \infty$ uses the complete hypernympaths of each ground-truth word to match the candidates.

4 Experiments

Initially, we discuss the relation extraction from recipes using BERT. This evaluation is based on the small set of manually labeled instructions. Subsequently, we analyze how well these extracted relations match the set of ground-truth relations in comparison to those of ConceptNet.

4.1 Relation Extraction From Recipes

We assess BERT's relation-extraction performance based on the set of 400 annotated instructions. We use a 5-fold cross validation with 50 repetitions. Figure 5 depicts on the left the learning curves of the model regarding the classification of single tokens as well as complete relations. The correct classification of a single relation is equivalent to perfectly classifying all tokens of the corresponding instruction. Hence, an accurate token classification is required to achieve reasonable results for whole relations, as can be seen by the discrepancy in the error rates. Even with 320 training examples a very low token-classification error is achieved (7.2 %) and around half the triplets are perfectly extracted (51.5 %). Figure 5 also illustrates the effectiveness of the post-processing (see Section 3.2) as it significantly improves the relation extraction. The curve seems already to converge after 100 training instructions. However, the total amount of correctly extracted relations is further increasing (Figure 5 on the right). In other words, the post-processing rejects fewer relations and the approach becomes more efficient, extracting more relations from the same amount of data.

The error of the relation classification after post-processing (brown curve) can be interpreted as an upper bound for the proportion of false relations. However, in practice the proportion is significantly lower as can be seen by our estimates in Section 4.3.1. The small dataset naturally leads to a high variance in the performance across different repetitions. It can be expected that the error is further reduced with additional supervised data as no saturation has been reached yet. In particular, considering the discrepancy between the language type used to pre-train BERT, proper English from books and Wikipedia articles, and the one of recipes, compressed short sentences often neglecting a valid grammar, the performance is likely to improve when this mismatch is further minimized.

4.1.1 Processing the Whole Dataset

BERT is fine-tuned with all annotated instructions to extract the relation set of Recipe1M+. Overall, the dataset contains around 10 million instructions of from which our pipeline extracts 28729 unique relations. The mean recurrence rate of a triplet is 4.4 with a median of 1. Table 2 lists the most frequent relations. Relations using a mixer / blender are predominant, which is reasonable as they are used in most baking recipes.

Table 2: The most frequent relations extracted from the recipes.

Relation	Recurrence
mixer, beat, butter	3491
mixer, beat, cheese	2052
mixer, beat, egg	1082
blender, cut, butter	931
mixer, cream, butter	889
rolling pin, roll, dough	783
mixer, beat, cream	704
blender, blend, ingredients	676
blender, puree, soup,	670
mixer, beat, ingredients	603

4.2 Relation Extraction from ConceptNet

We briefly describe the straight-forward extraction from ConceptNet. Starting from our set of predefined tools we use only the relevant link-types "used for" and "capable of" to extract the relations. These link types connect the tools with single words or short phrases. We use the syntactic parsing of spaCy (Honnibal and Montani, 2017) to extract the action and object from the short phrases based on a few case-based rules. ConceptNet contains 2574 entries for our tool set and the considered link-types. However, such entries often lack an object as required for the type of relations we are aiming for, e. g. `knife-used for-cutting", "fork-used for-eating". Alltogether, we extracted 1322 complete relations that are in accordance with the WordNet vocabulary.

4.3 Relation Matching

We determine the matching rate between the ground-truth relations from the study and both extracted sets respectively. The rate is measured as defined by Equation 2. Figure 6 depicts how the matching improves when the maximum distance threshold k is increased. The recipe relations match distinctly more of the ground-truth relations. Concretely, they yield three times more "perfect" matches ($k = 0$). The relations of ConceptNet profit more from an increasing distance threshold. The probable explanation is that recipes usually use very specific terms to precisely describe the single steps, whereas ConceptNet contains information concerning more general terms that are more likely to match for larger distance thresholds. This hypothesis is supported by the fact that the average length of the hypernympaths assigned to the synsets within the ConceptNet triplets is smaller than those of the recipes (6.1 vs. 7.2). The structured data of ConceptNet facilitates the relation extraction,

Figure 5: On the left: Learning curves of the token- and relation classification. The relation classification is depicted before (solid blue) and after post-processing(dashed brown).
On the right: Proportion of relations that are correctly extracted after post-processing.

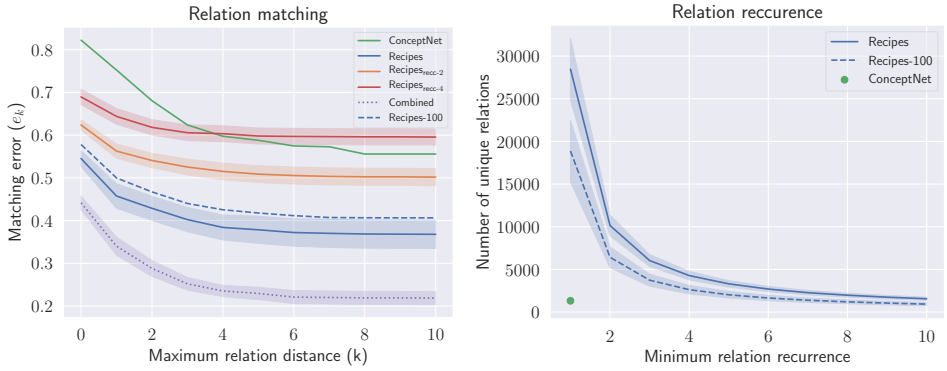


Figure 6: On the left: Matching error rate e_k for an increasing distance threshold k. The advantage of the extraction from recipes is particularly pronounced for exact matches (k=0). On the right: Amount of extracted unique relations when the minimum recurrence is varied.

leading to a naturally low rate of false relations, whereas the opposite applies for the extraction from recipes. However, the recurrence of the recipe relations can be used as confidence for their validity, providing a way to control the proportion of false relations against the number of extracted relations. Therefore, we also illustrate the performance for a minimum recurrence rate of two and four.

To assess the minimum amount of required labeled instructions, we trained another BERT model based using only 100 training instructions ("Recipe-100" in Figure 6). Its matching error is only slightly worse in comparison to the model using 400 labeled instructions, suggesting that even fewer examples may be sufficient to achieve comparable results.

Figure 6 depicts on the right the amount of unique relations amount depending on the minimum recurrence rate. Even a minimum recurrence rate of 10 yields more unique relations than ConceptNet. This graph points out the massive discrepancy in the amount of the relations yielded by BERT over those contained in ConceptNet. The BERT model trained with 100 examples extracts distinctly fewer relations, which is in line with the right plot of Figure 5 and confirms that more training examples in particular increase the data efficiency of the model. It is not surprising that the combination of both sets leads to the overall the best-performance as shown by the purple curve in Figure 6. However, it is noteworthy that the relations are complementary to some degree, since the improvement is significant ($> 10\,\%$), suggesting the fusion of both approaches.

4.3.1 Taking False Relations into Account

The correctness of common-sense relations is of utmost importance. In case of planning algorithms, false relations can prevent the generation of a plan or even result in incorrect ones, potentially leading to severe failures. In our case, false relations are

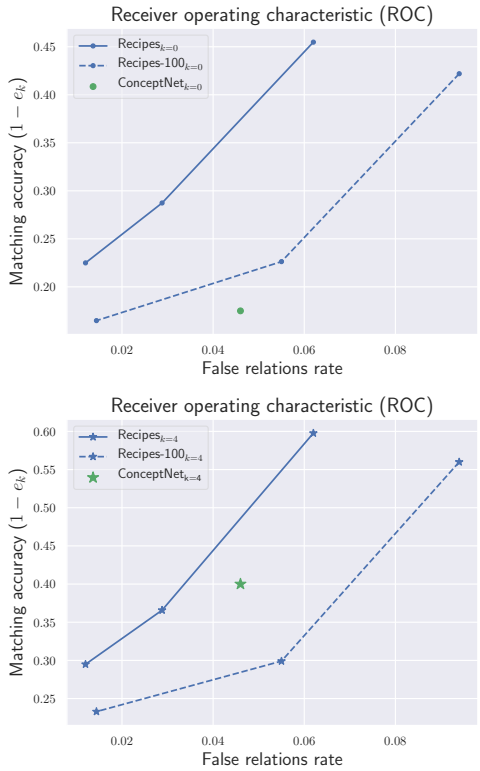


Figure 7: ROC curves for exact matches ($k = 0$, on the left) and for matches with a maximum distance of $k = 4$ (on the right). The false relation rates were determined based on sampling and manual inspection. The rates of the recipes are varied by adjusting the minimum recurrence rate.

mainly caused by the extraction process, as the relations in the recipes as well as ConceptNet generally are valid in some context.

We estimate the false relation rate based on multiple samples, thereby manually inspecting ~ 3000 relations[1]. Figure 7 shows the resulting ROC curves. Concretely, it provides ROC curves for maximum relation distances of $k \in \{0, 4\}$. In case of the recipes, we control the proportion of false relations by varying the minimum recurrence. We sampled the false relations rate for reccurence rates of $\{1, 5, 9\}$. The results of ConceptNet are represented by single points, since recurrence-based filtering is not applicable for its unique relations. The false relation rate of the recipes is reduced for higher recurrence rates. The corresponding curves yield superior results in compared to the values of ConceptNet, particularly for exact matches ($k = 0$). To put it in a nutshell, the relations extracted from recipes do not only match distinctly more relations that are naturally named by humans, but also yield a lower rate of false relations when the minimum recurrence is accordingly adjusted.

The analysis may seem to be biased, since we compare the relations of a general-purpose database with those of domain-specific procedural task instructions. Particularly, considering the fact that the kitchen domain is very popular with an abundance of publicly available data. This is a valid point and we are currently considering a comparison to sources providing procedural task instructions for a broad range of tasks such as wikiHow. However, our main point is not to stress the fact that more relations can be extracted from procedural task instructions. Instead, we demonstrate that with a relatively small effort BERT can be trained to extract these relations with a high precision leading to overall superior results.

5 Conclusion

We explored whether BERT can be used to extract common-sense relations from procedural task instructions as an alternative to querying public databases. We fine-tuned BERT for the relation extraction from recipes based on very few labeled instructions and extracted the relations from the large *Recipe1M+* dataset. To assess their relevance we collected a set of ground-truth relations in a study and proposed an evaluation measure that utilizes the WordNet hypernym graph to incorporate matches between specific and general terms. The matching granularity can naturally be adjusted, allowing a diverse analysis. The experiments highlight various advantages of the BERT based approach. It does not only yield a very large amount of unique relations (28k versus 1.3k) and correspondingly matches a large portion of the ground-truth relations, but the recurrence of the relations can also be used to reduce the proportion of false relations. Therefore, we regard the extraction of common-sense relations from text as a competitive and complementary approach, particularly considering the ongoing and rapid advance of NLP techniques.

[1]The sample size for a confidence width of $w = 0.04$ is determined by the number of false relations within an initial sample of 100 relations.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley FrameNet project. In *17th International Conference on Computational Linguistics*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *SemEval@ACL*, pages 15–26.

Joyce Y. Chai. 2018. Language to action: Towards interactive task learning with physical agents. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18. International Foundation for Autonomous Agents and Multiagent Systems.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Qiaozi Gao, Shaohua Yang, Joyce Chai, and Lucy Vanderwende. 2018. What action causes this? Towards naive physical action-effect prediction. In *ACL (1)*, pages 934–945.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, volume 1, page 5.

David Gunning. 2018. Machine common sense concept paper. *arXiv:1810.07528*.

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

Yidan Hu, Gongqi Lin, Yuan Miao, and Chunyan Miao. 2019. Commonsense knowledge+ bert for level 2 reading comprehension ability test. *arXiv preprint arXiv:1909.03415*.

Soufian Jebbara, Valerio Basile, Elena Cabrio, and Philipp Cimiano. 2019. Extracting common sense knowledge via triple ranking using supervised and unsupervised distributional models. *Semantic Web*, 10(1):139–158.

Peter Kaiser, Mike Lewis, Ronald P. A. Petrick, Tamim Asfour, and Mark Steedman. 2014. Extracting common sense knowledge from text for robot planning. In *ICRA*, pages 3749–3756.

Mahnaz Koupaee and William Wang. 2018. Wikihow: A large scale text summarization dataset. In *arXiv:1810.09305*.

Jens Lehmann, Robert Isele, Jakob, et al. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *ACL (1)*.

Vincent C Müller and Nick Bostrom. 2016. Future progress in artificial intelligence: A survey of expert opinion. In *Fundamental issues of artificial intelligence*, pages 555–572.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *International Conference on World Wide Web*, pages 697–706.

Niket Tandon, Gerard De Melo, and Gerhard Weikum. 2017. Webchild 2.0: Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, pages 115–120.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Yezhou Yang, Anupam Guha, Cornelia Fermüller, and Yiannis Aloimonos. 2014. Manipulation action tree bank: A knowledge resource for humanoids. In *Humanoids*, pages 987–992. IEEE.

Shaohua Yang, Qiaozi Gao, Changsong Liu, Caiming Xiong, Song-Chun Zhu, and Joyce Chai. 2016. Grounded semantic role labeling. In *HLT-NAACL*, pages 149–159.

Yiben Yang, Larry Birnbaum, Ji-Ping Wang, and Doug Downey. 2018. Extracting commonsense properties from embeddings with limited human guidance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 644–649.

Philipp Zech, Erwan Renaudo, Simon Haller, Xiang Zhang, and Justus Piater. 2019. Action representations in robotics: A taxonomy and systematic classification. *International Journal of Robotics Research*, 38(5):518–562.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. " going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. *arXiv preprint arXiv:1909.03065*.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. *arXiv preprint arXiv:2005.04304*.