The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Recognizing and Justifying Text Entailment through Distributional Navigation on Definition Graphs

Vivian S. Silva,[1] **André Freitas,**[2] **Siegfried Handschuh**[1]
[1]Department of Computer Science and Mathematics, University of Passau, Innstraße 43, 94032, Passau, Germany
[2]School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, M13 9PL, UK
vivian.santossilva@uni-passau.de, andre.freitas@manchester.ac.uk, siegfried.handschuh@uni-passau.de

Abstract

Text entailment, the task of determining whether a piece of text logically follows from another piece of text, has become an important component for many natural language processing tasks, such as question answering and information retrieval. For entailments requiring world knowledge, most systems still work as a "black box", providing a yes/no answer that doesn't explain the reasoning behind it. We propose an interpretable text entailment approach that, given a structured definition graph, uses a navigation algorithm based on distributional semantic models to find a path in the graph which links text and hypothesis. If such path is found, it is used to provide a human-readable justification explaining why the entailment holds. Experiments show that the proposed approach present results comparable to some well-established entailment algorithms, while also meeting Explainable AI requirements, supplying clear explanations which allow the inference model interpretation.

Introduction

Natural Language Processing tasks such as question answering, text summarization and information retrieval often rely on text entailment as a means of identifying and interpreting semantic relationships between pieces of text. Text entailment is defined as a directional relationship between a pair of text expressions, denoted by T – the entailing text, and H – the entailed hypothesis. We say that T entails H if, typically, a human reading T would infer that H is most likely true (Dagan, Glickman, and Magnini 2006). Although a human could easily explain why they consider an entailment true, most text entailment systems still can't provide a comprehensible justification for their decisions, because this usually depends on knowledge that goes beyond what is stated in the text and hypothesis. Given the growing importance of Explainable AI (Gunning 2017), systems can no longer omit the reasons why decisions are reached, making justifications a fundamental feature for intelligent systems.

Recent text entailment approaches, especially those relying on more complex semantic interpretation, use knowledge bases and linguistic resources to track down semantic relationships between text and hypothesis. WordNet (Fellbaum 1998) is notably the most commonly used resource, but systems usually exploit only the links between terms, such as synonym, hypernym or derivational form relationships (Clark, Fellbaum, and Hobbs 2008; Herrera, Penas, and Verdejo 2006). The term's definition (i.e. the gloss), which contains the largest bulk of relevant information about it, is left aside.

In this work, we propose an approach for recognizing text entailments that uses knowledge extracted from natural language lexical definitions, structuring them into a semantic representation model that allows the identification of semantic relationships between terms. We focus on text entailments that require reasoning over world knowledge, and use a navigation algorithm based on distributional semantics to scope the exploration of a knowledge graph built from dictionary definitions, looking for paths between the text and hypothesis, confirming or rejecting the entailment.

The main contribution of our approach is to provide an interpretable reasoning model for text entailment by automatically building commonsense knowledge bases out of natural language definitions. This model was designed to meet Explainable AI requirements: besides giving a yes or no answer, the resulting path provides a justification about the entailment decision, that is, a human-readable explanation exposing the reasoning steps that led to the answer is offered to support it, allowing users to be aware of, evaluate and judge the inference model employed in the task.

Text Entailment

Stimulated by the RTE (Recognizing Text Entailments) Challenges[1], a large number of text entailment frameworks have been developed in the last years. Starting in 2005, the RTE Challenges encouraged the creation of systems capable of capturing semantic inferences, and, given the low accuracy achieved by the first participants, showed that much improvement was still required in the area (Ghuge and Bhattacharya 2014).

Over the last editions, the RTE Challenges have moved from shallow methods, based on lexical features, to more sophisticated approaches, called deep methods, which rely on semantic, syntactic and/or logical features. Some RTE-1 participants presented systems based purely on word overlap and statistical lexical relations (Glickman and Dagan 2005;

[1]https://goo.gl/R9zVqp

Pérez and Alfonseca 2005; Newman et al. 2005), while subsequent editions of the challenge introduced approaches using logical inference, machine learning and linguistic resources such as WordNet, Framenet and Verbnet, among other techniques, as features in the text entailment recognition process. As a common starting point, those approaches translate the text and hypothesis to some kind of (syntactic or semantic) representation, and then try to determine if the representation of the hypothesis is subsumed by that of the text.

Most recent text entailment approaches are machine learning-based (Kouylekov and Magnini 2005; Wang and Neumann 2008; Zhang et al. 2017), combining multiple similarity measures (computed over lexical, syntactic and semantic representations) to train a machine learning model. Entailment pairs [T, H] are represented as feature vectors $\{f_1, f_2, ..., f_m\}$, which are manually classified as entailment/non-entailment, feeding a supervised machine learning model. The features can range from similarity measures applied to the pair to the sentence's syntactic or semantic representations, such as their parse trees or semantic relations graph. After training, the model can then classify unseen entailment pairs by examining their features (Androutsopoulos and Malakasiotis 2010).

Most entailment systems provide as output a yes/no answer and a confidence score, but no justification or evidence that support the entailment decision. A few exceptions include the Boeing Language Understanding Engine (BLUE) (Clark and Harrison 2009), which can show evidence of why an entailment was achieved, but doesn't provide a fully interpretable natural language explanation. The textual inference approach proposed by (Raina, Ng, and Manning 2005) also provides a kind of justification for entailments, through a logical theorem prover that outputs a minimum cost proof, which can, in turn, be translated into a natural language explanation for the inference.

The Third RTE Challenge proposed an optional task which required a system to make three-way entailment decisions (entails, contradicts, neither) and to justify its response. Analyzing the outputs provided by the competing systems, they point that explanations for why the hypothesis is entailed widely differ, however, with some rationales of dubious validity (Voorhees 2008). Human evaluators listed a number of problems, among which is worth mentioning the use of vague and abstract phrases such as "there is a relation between" and "there is a match", showing that describing the specific semantic relation is fundamental to build the trust in the system and in its reasoning methods (rather than simply detecting that the semantic relation exists).

Distributional Navigation on Definition Graphs

The approach proposed in this work is based on two main pillars: the use of a knowledge graph automatically extracted from natural language definitions as world knowledge base, and a navigation mechanism based on distributional semantics to explore this graph and find paths between the text and the hypothesis to explain the semantic relationships holding between them, which confirm and support the entailment. The proposed method contributes in two directions: (i) providing an interpretable text entailment approach which provides justifications and (ii) defining a method for building world KBs out of definitions expressed in natural language. The definition graph construction and the distributional navigation algorithm are described in the next Sections.

Definition Graph

The graph used as knowledge base is composed of lexical definitions and is built from a linguistic resource following the representation model proposed by (Silva, Handschuh, and Freitas 2016). In this model, the definitions are split into entity-centered *semantic roles*, which express the part played by an expression in a definition, showing how it relates to the entity being defined. Table 1 lists the semantic roles for lexical definitions present in the model, and Figure 1 shows an example of a concept (in this case, a synset from WordNet) classified according to the model.

Table 1: Semantic roles for dictionary definitions.

Role	Description
Supertype	the immediate or ancestral entity's superclass
Differentia quality	a quality that distinguishes the entity from the others under the same supertype
Differentia event	an event (action, state or process) in which the entity participates and that is mandatory to distinguish it from the others under the same supertype
Event location	the location of a differentia event
Event time	the time in which a differentia event happens
Origin location	the entity's location of origin
Quality modifier	degree, frequency or manner modifiers that constrain a differentia quality
Purpose	the main goal of the entity's existence or occurrence
Associated fact	a fact whose occurrence is/was linked to the entity's existence or occurrence
Accessory determiner	a determiner expression that doesn't constrain the supertype-differentia scope
Accessory quality	a quality that is not essential to characterize the entity
[*Role*] particle	a particle, such as a phrasal verb complement, non-contiguous to the other role components

This model allows a structured semantic representation of natural language definitions and enables the selection of the portions of information that are relevant for a given reasoning task. To use lexical definitions as world knowledge in

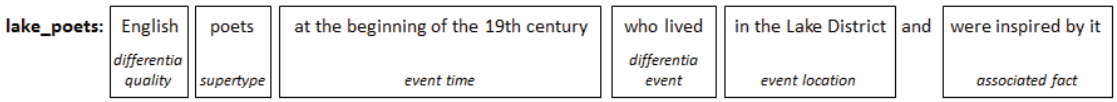

Figure 1: Example of role labeling for the definition of the "lake_poets" synset.

our text entailment approach, we used WordNet definitions, identifying their semantic roles and then converting the resulting semantic model to an RDF graph.

To identify the semantic roles, we first automatically pre-annotated a random set of 2,000 WordNet noun and verb definitions. The rule-based automatic annotation used the syntactic patterns identified by statistical analysis as described by (Silva, Handschuh, and Freitas 2016): after generating the syntactic parse tree for each definition, using a C-Structure parser (Manning et al. 2014), the relevant phrasal nodes were identified and the semantic role more often associated to it was assigned to the definition's segment represented by this syntactic structure.

After the automatic annotation, the set of 2,000 definitions was then manually curated in order to fix misclassifications and fill in missing roles. The curated data was then used to train a Recurrent Neural Network (RNN) machine learning model designed for sequence labeling. We used the RNN implementation provided by (Mesnil et al. 2015), and split the data into training (68%), validation (17%) and test (15%) sets. The trained classifier reached an accuracy of 80.35%. We then used it to classify all the WordNet's noun and verb definitions.

After classifying the definitions, an RDF graph representation was generated, where each synset (a set of synonym words) in WordNet is a node, and each segment in its definition is another node, linked to the synset node and among them by the properties given by the roles. In this graph, called WordNetGraph[2], the *definiendum*, i.e., the synset, is linked to its supertype, which is, in turn, linked to all the other roles. A role is a resource whenever it is linked to other roles, and a literal otherwise. Figure 2 shows the (simplified) RDF representation of the definition depicted in Figure 1.

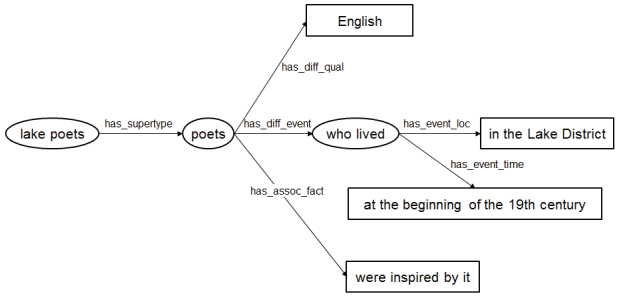

Figure 2: RDF representation for the definition of the "lake poets" synset.

As can be seen, the definiendum "lake poets" is linked to its supertype "poets", which is linked to the other roles. The entity "poet" has its own definition, which is represented as another subgraph, making the whole graph interconnected through the words that appear in each definition.

We assume that, when an entailment is true, the core words present in the text and hypothesis have a strong semantic relationship, and then it is possible to find a path in the definition graph linking them and justifying the entailment. To find this path, we navigate in the graph using the algorithm described in the next Section.

Distributional Navigation

Distributional Semantic Models (DSMs) are grounded in the distributional hypothesis, which states that words that occur in similar contexts tend to have similar meanings (Turney and Pantel 2010). DSMs allow the approximation of a word meaning representing it as a vector summarizing its pattern of co-occurrence in large text corpora (Marelli et al. 2014).

DSMs can be used to compute the *semantic similarity/relatedness measure* between words. This computation is used as a heuristic to navigate in a graph knowledge base in the approach proposed by (Freitas et al. 2014), where they define a *Distributional Navigation Algorithm* (DNA), which corresponds to a selective reasoning process in the knowledge graph. Given a pair of terms, namely a *source* and a *target*, and a threshold η, the DNA finds all paths from *source* to *target*, with length l, formed by concepts semantically related to *target* wrt η (Freitas et al. 2014).

In the text entailment context, the source and target terms are words from the text and hypothesis, respectively, which have some kind of semantic relationship between them. A path in the definition graph linking these terms, then, explains what this relationship is, confirming the entailment, or rejecting it in case no path is found.

We implement the DNA as a *depth first search* algorithm, exploring first the paths whose next node to be visited has the highest semantic similarity value wrt the target. Given a node n in the knowledge graph, starting from the source, the algorithm retrieves all its neighbors $\{t_1, t_2, ..., t_n\}$ and computes the similarity relatedness $sr(t_i, target)$, keeping only the nodes for which $sr > \eta$ in the set of nodes to be visited next. Each of these nodes generates a new path, and, for each path, the search goes on until the next node to be visited is equal to the target, or until the maximum path length is reached. If no path reaches the target before the maximum number of paths is reached, the search stops.

The search algorithm is listed below:

Input:
- definition graph G
- source word S

[2]https://github.com/Lambda-3/WordnetGraph

- target word T
- threshold η
- path length l
- max number of paths m

Output:
A set of paths from S to T

```
paths = []
stack = []
new_path = [S]
stack.push(new_path)

while (stack is not empty and paths.size < m)
    path = stack.pop()
    next_node = last_node(path)
    while (next_node ≠ target and path.length < l)
      get all the synsets {s1, s2, ..., sn} in G having next_node
      nodes = []
      best_roles = []
      for each si in {s1, s2, ..., sn}
        get all the role nodes {r1, r2, ..., rn} linked to si
        for each ri in {r1, r2, ..., rn}
          if sr(ri, T) > η
            nodes.add(ri)
      best_roles = sort(nodes)
      nodes = []
      ranked_head_words = []
      for each bi in best_roles {b1, b2, ..., bn}
        get all the head words {h1, h2, ..., hn} in bi
        for each head word hi in {h1, h2, ..., hn}
          nodes.add(hi)
      ranked_head_words = sort(nodes)
      for each wi in ranked_head_words {w2, w3, ..., wn}
        new_path = path
        new_path.add(wi)
        stack.push(new_path)

      next_node = w1
      path.add(next_node)
      if (next_node = target)
        paths.add(path)
```

As can be noted in the algorithm, the next node to be visited in a path is defined by the words present in a synset's definition, here called *head words*. The head words are the most relevant words in a role, and are identified following a lexico-syntactic rule-based heuristic: it is usually a noun for non-event-centered roles such as supertype and differentia quality, for example, and the main verb along with its noun complements or adjective/adverb modifiers in case no relevant noun is present, for event-centered roles, such as differentia event, associated fact or purpose, for instance.

According to (Freitas et al. 2014), the worst-case time complexity of the DNA implemented as a depth-first search "is $O(b^l)$, where b is the branching factor and l is the depth limit". They show that the selectivity of DNA ensures that the number of paths does not grow exponentially even when the depth limit increases. In our implementation, the algorithm parameters - threshold, maximum number of paths and maximum path length (depth limit) - were obtained empirically in order to optimize the search.

Recognizing and Justifying Text Entailments

Once we have a graph knowledge base and a method to navigate over this graph, we can use these resources to compose the reasoning mechanism that will allow us to recognize and explain a text entailment. We are interested in entailments that require world knowledge, over which some kind of inference is necessary, rather than simple syntactic variations between the text and the hypothesis. For example, consider the following entailment pair from the Boeing-Princeton-ISI (BPI)[3] dataset:

64.2 T: Skilling was wearing a security tag on his ankle when he stepped into the street to face the press.
64.2 H: Skilling was wearing a security tag.

In this example, the hypothesis is fully contained in the text, and no knowledge external to the entailment pair is necessary, therefore no actual semantic reasoning is required. On the other hand, in the following example, also from the BPI dataset, a simple syntactic analysis would not suffice:

39.3 T: Many cellphones have built-in digital cameras.
39.3 H: Many cellphones can take pictures.

In this case, it is necessary to answer a question: "Given that cellphones have digital cameras, is it true that they can take pictures?". What we propose is to look for the answer to this question looking at the structured definitions in our knowledge graph to check whether the hypothesis is reached from the text in some way. If so, the way this link is established gives a full answer to the original question.

First, we need to identify the relevant elements from the text and hypothesis for which is worth to look for a semantic relationship. If the text is too long and includes more than one clause, we perform a sentence simplification to break it into independent simple sentences, and then choose among them the one that is closest to the hypothesis, using simple Levenshtein edit distance. The edit distance proved to be sufficient at this step, as we just want to identify what text sentence refers to the same topic as the hypothesis, and so share more elements with it. Consider as example the following BPI entailment pair:

3.6 T: Hanssen, who sold FBI secrets to the Russians, could face the death penalty.
3.6 H: Hanssen received money from the Russians.

After the sentence simplification, the text is split into two sentences: "Hanssen could face the death penalty" and "Hanssen sold FBI secrets to the Russians". The second one is the closest to the hypothesis and is selected to compose the new entailment pair.

[3] http://www.cs.utexas.edu/users/pclark/bpi-test-suite/

Next, we look for the *core words* in the text and hypothesis. The core words are similar to the head words for definition's roles, but in this case we have full sentences rather than sentence segments, as happens with the roles, so here we can perform a more accurate syntactic analysis. Also following a rule-based heuristic, we get the main noun in the subject, the main verb and its noun complements, or the adjective/adverb modifiers in case no relevant noun is found. Back to the pair 39.3, the core words for the text "Many cellphones have built-in digital cameras" are *cellphones*, *have* and *digital cameras*; and for the hypothesis "Many cellphones can take pictures", *cellphones*, *take* and *pictures*.

We then discard the overlapping words and words with low *inverse document frequency* (IDF), which are words that are too frequent and can be reached from almost any node in the graph, leading to diverting paths, such as the verbs *get*, *put*, *cause* or *make*, to name a few. IDF is calculated using the set of all definitions in WordNet as the corpus, where each definition is considered a document. Next, we normalize all the remaining words, obtaining two resulting sets of core words, $C_T=\{t_1, t_2, ..., t_n\}$ for the text and $C_H=\{h_1, h_2, ..., h_m\}$ for the hypothesis. Also using distributional semantics, we compute the semantic similarity measures between all the core words, as a Cartesian product between C_T and C_H. The results are sorted and the *k* pairs with the highest similarity values are chosen, being $k=max(n, m)$, where *n* is the size of C_T and *m* is the size of C_H. Since each pair is composed of a word (or phrase) from the text and another from the hypothesis, these will be the input for the navigation algorithm described earlier.

For each pair of words found in the previous step, we find all the paths between them in the definition graph, the source being the word from the text, and the target the word from the hypothesis. Finally, we choose, among all the paths found, the smallest one, which is the one that offers the shortest distance between a source and a target and, therefore, shows that their meanings are more closely related.

The final path is composed of a sequence of synset nodes and the role nodes that make up those synset's definitions and that are relevant to build a composed relationship between the source and the target. This sequence of nodes is then formatted to provide a human-readable justification explaining the reasoning that led from the text to the hypothesis, giving the necessary evidence that the latter logically follows from the former.

Figure 3 shows an example of a path in the WordNetGraph between the source "digital camera" and the target "picture", from the entailment pair 39.3. Starting from the source node, we get all the nodes linked to it, compute their semantic similarity measures wrt the target, choose the node with the highest value as the next one to be visited, and do this recursively until we reach the target. Other nodes with high similarity values (higher than the threshold), such as the differentia quality node "that encodes an image digitally", are also explored later, but the path indicated by the thicker lines in the figure is the shortest, and therefore the best, one.

Despite the low similarity value, the supertype node "equipment" is included in the path in order to provide the necessary information for the justification. The justification takes into account the content of the nodes and the relationships between them, that is, the role names, and a differentia quality (as well as almost all the other roles) doesn't make much sense without the supertype it is linked to. For this example, the final, human-readable explanation generated by the algorithm from this sequence of nodes is:

A digital camera is a kind of camera
A camera is an equipment for taking photographs
Photograph is synonym of picture

Evaluation

To evaluate the proposed approach, we run experiments using the BPI dataset and a sample of the Guardian Headlines dataset[4]. These showed to be the most suitable datasets, since RTE ones focus more on linguistic phenomena. The BPI dataset focus more on the knowledge necessary to recognize the entailments rather than just linguistic requirements, being syntactically simpler than RTE datasets, but more challenging from the semantic viewpoint. It is composed of 250 entailment pairs, being 125 positive and 125 negative entailments. The Guardian Headlines dataset is a set of 32,000 entailment pairs automatically extracted from The Guardian newspaper. Its large size is intended for machine learning purposes, but, as it wasn't validated, a manual curation is necessary, which is possible only for a sample of the data. We randomly selected 10% of the pairs (3,200), and from this sample we selected 800 pairs, in order to make it close in size to RTE datasets. We excluded pairs where there was a pronoun referencing an entity not present in the sentence (for negative examples) and pairs where the hypothesis contained a question, a named entity not contained in the text, or had information that was more specific than that expressed by the text (for positive examples). The resulting dataset[5], herein called Guardian Headlines Sample (GHS), has 400 positive and 400 negative pairs, covering a wide range of entailment phenomena, while still requiring a reasonable amount of world knowledge.

Along with the WordNetGraph, we used word2vec (Mikolov et al. 2013) as the distributional semantic model to carry out the distributional navigation on the knowledge graph. We used the Indra[6] (Freitas et al. 2016) service to compute the semantic similarity measures. For simplifying long text sentences, we used the Sentence Simplification service (Niklaus et al. 2016) in the information extraction pipeline Graphene[7].

We compare the results with two baselines generated by the *Excitement Open Platform* (EOP) (Magnini et al. 2014), a framework that implements state-of-the-art text entailment algorithms: *tree edit-distance based* and *classification based*, through the *EditDistance* (Kouylekov and Magnini 2005) and *MaxEntClassification (Base+WN+TP+TPPos+TS_EN)* (Wang and Neumann 2008) implementations, respectively. The default settings

[4]https://goo.gl/XrEwG9

[5]The GHS curated dataset is available at https://goo.gl/4iHdbX

[6]https://github.com/Lambda-3/Indra

[7]https://github.com/Lambda-3/Graphene

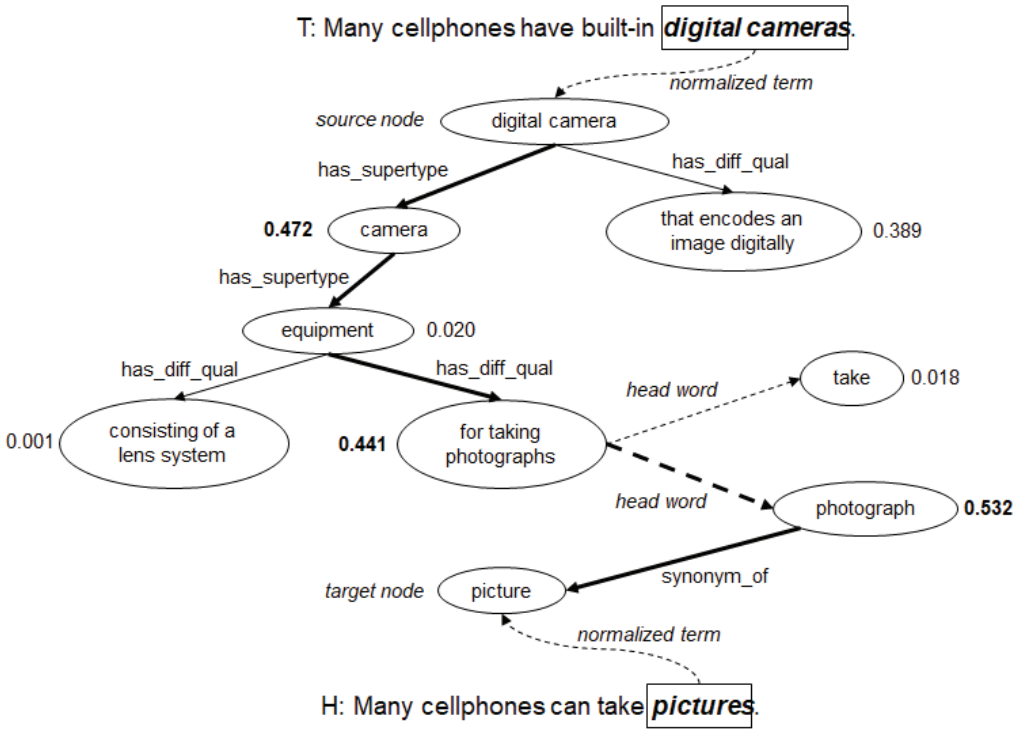

Figure 3: Possible paths in WordNetGraph between "digital camera" and "picture". Full lines represent actual edges in the graph, while dashed lines represent the navigation algorithm's internal operations. The best path is indicated by the ticker lines. Numbers show the semantic relatedness between each node and the target.

were kept and the models were trained following the EOP documentation instructions[8].

The precision, recall and F-measure obtained by the graph navigation algorithm, as well as the baselines, are presented in Table 2. Even though our approach is focused on a specific scenario, the results are comparable to those of the complete entailment systems.

Table 2: Evaluation results. The upper part shows the baselines, and at the bottom are the proposed graph navigation approach's results.

	BPI			GHS		
	Pr	**Re**	**F1**	**Pr**	**Re**	**F1**
EditDist	0.44	0.65	0.53	0.96	0.30	0.45
MaxEnt	0.46	0.57	0.51	0.50	1.00	0.66
GraphNav	0.65	0.54	0.59	0.56	0.50	0.53

The proposed approach presents slightly better results for the BPI dataset, since it favors the world knowledge exploration. The GHS is a challenging dataset since it contains longer and more complex sentences, and frequently shows substantial vocabulary variation between text (the first line of a story) and hypothesis (the story's headline), given that journalists tend to avoid repetition of words. The EditDistance algorithm shows high precision on this dataset, possibly because it has no "tricky" negative examples from the syntactic point of view, as BPI does, but presents low recall. The MaxEntClassification algorithm surprisingly classifies all but two pairs in GHS as entailment, reaching the maximum recall but, since the dataset is balanced, only half of the precision. Although our results are only comparable, our main contribution, in addition to the fine-grained semantic matching, are the justifications generated by the navigation algorithm which make the results interpretable. In the next Section, we do a qualitative analysis of the justifications.

[8]https://goo.gl/aFyoCh

Justifications

The justifications generated for the positive entailments were manually evaluated by two independent human evaluators in order to assess their correctness and consistency. Three types of justification were noticed: *correct and complete*, *correct but incomplete* and *incorrect*.

Correct and complete: the reasoning is clear and the important information that explains the entailment is present in the justification. Some examples from the BPI dataset:

57.1 T: Many soldiers were killed in the ambush.
57.1 H: The soldiers were attacked by surprise.
57.1 A: YES

Entailment: Yes
Justification:

An ambush is an act of concealing yourself and lying in wait to attack by surprise

65.2 T: Sony apologized Tuesday for inconvenience caused by a global recall of laptop batteries.
65.2 H: The batteries were defective.
65.2 A: YES

Entailment: Yes
Justification:
A recall is a request by the manufacturer of a defective product to return the product

From the GHS dataset:

21660 T: Cadbury Schweppes plc today said ahead of its annual general meeting that it was on course to hit sales targets and its US businesses were flourishing.
21660 H: Cadbury set to meet sales targets
21660 A: YES

Entailment: Yes
Justification:
To hit is a way of to encounter
To encounter is synonym of to meet

29121 T: A former security guard at Michael Jackson's Neverland ranch testified yesterday that he had seen the pop star perform oral sex on a young boy in the early 1990s.
29121 H: Court hears Jackson oral sex claim
29121 A: YES

Entailment: Yes
Justification:
To testify is to give testimony in a court of law
Court of law is synonym of court

Correct but incomplete: although the information contained in the justification is correct, it explains the reasoning only partially, not covering all semantic relations between text and hypothesis. Example:

29.3 T: Foodstuffs are being blocked from entry into Iraq.
29.3 H: Food cannot get into Iraq.
29.3 A: YES

Entailment: Yes
Justification:
A foodstuff is a substance that can be used or prepared for use as food

Incorrect: the justification is too vague or doesn't establish the correct link between text and hypothesis. Example:

13468 T: The BBC today beat stiff competition from ITV to secure coverage of the Grand National until the end of the decade.
13468 H: BBC wins race for Grand National
13468 A: YES

Entailment: Yes
Justification:
A competition is an occasion on which a winner is selected from among two or more contestants
A winner is a contestant who wins the contest

Incorrect justifications are generated due to a wrong choice of source/target word pairs. As pointed before, the main goal of our approach is to recognize and explain text entailments where world knowledge plays a central role and the hypothesis is not a syntactic variation of the text, but rather a statement that holds a semantic relationship with it. Pairs where all the words in the hypothesis are also present in the text are particularly challenging, since, in this case, it is hard to find a suitable pair of words to be sent as input to the distributional navigation algorithm, requiring a complementary strategy to tackle those scenarios.

When the sentences remain complex even after the simplification, as sometimes is the case in the GHS dataset, selecting the core words can also be challenging, leading to decisive source/target pairs being missed. Fine tuning the core words identification to address more elaborate sentence structures could help to fix this kind of loss.

Finally, misclassification or the absence of relevant information in the definition graph also leads to incorrect entailment decisions. Misclassifications account mainly for syntactic errors in the justifications, while the lack of information prevents the navigation algorithm from finding a path when a relationship between the source and target indeed exists, rejecting a true entailment and reducing the accuracy. A possible solution is to enrich the knowledge graph with definitions from other linguistic resources, such as Wiktionary.

Conclusion

We presented an approach for recognizing and justifying text entailments that require reasoning over world knowledge. Using lexical definitions as a knowledge base, we built a knowledge graph and implemented a distributional navigation algorithm to explore it. Words from the text and hypothesis having a strong semantic relationship are chosen as source and target pairs, and all the paths between them in the knowledge graph are retrieved. The shortest path is chosen and interpreted to generate a human-readable justification explaining the reasoning behind the entailment decision. If no path is found, then the entailment is rejected.

The major contribution of our approach is to provide a way to interpret and understand the underlying inference model, making the information used clear and expressing all the reasoning steps in a human-like manner, taking the entailment decision out of the numerical score black box. Another contribution worth mentioning is the transportability to other domains, since many fields have natural language glossaries but no structured thesauri. As future work, we intend to enhance the navigation algorithm to address a wider range of scenarios, improving its accuracy, and to enrich the knowledge graph with definitions from different linguistic resources to generate even higher quality justifications.

Acknowledgments

Vivian S. Silva is a CNPq Fellow – Brazil.

References

Androutsopoulos, I., and Malakasiotis, P. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research* 38:135–187.

Clark, P., and Harrison, P. 2009. An inference-based approach to recognizing entailment. In *TAC*.

Clark, P.; Fellbaum, C.; and Hobbs, J. 2008. Using and extending WordNet to support question-answering. In *Proceedings of the 4th Global WordNet Conference (GWC'08)*.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges: evaluating predictive uncertainty, visual object classification, and recognising textual entailment*. Springer. 177–190.

Fellbaum, C. 1998. *WordNet*. Wiley Online Library.

Freitas, A.; da Silva, J. C. P.; Curry, E.; and Buitelaar, P. 2014. A distributional semantics approach for selective reasoning on commonsense graph knowledge bases. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, 21–32. Springer.

Freitas, A.; Barzegar, S.; Sales, J. E.; Handschuh, S.; and Davis, B. 2016. Semantic relatedness for all (languages): A comparative analysis of multilingual semantic relatedness using machine translation. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, 212–222. Springer.

Ghuge, S., and Bhattacharya, A. 2014. Survey in textual entailment. *Center for Indian Language Technology*.

Glickman, O., and Dagan, I. 2005. A probabilistic setting and lexical cooccurrence model for textual entailment. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 43–48. Association for Computational Linguistics.

Gunning, D. 2017. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*.

Herrera, J.; Penas, A.; and Verdejo, F. 2006. Textual entailment recognition based on dependency analysis and WordNet. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*. Springer. 231–239.

Kouylekov, M., and Magnini, B. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, 17–20.

Magnini, B.; Zanoli, R.; Dagan, I.; Eichler, K.; Neumann, G.; Noh, T.-G.; Pado, S.; Stern, A.; and Levy, O. 2014. The excitement open platform for textual inferences. In *ACL (System Demonstrations)*, 43–48.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford Corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, 55–60.

Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; and Zamparelli, R. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*, 216–223.

Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23(3):530–539.

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*, 3111–3119.

Newman, E.; Stokes, N.; Dunnion, J.; and Carthy, J. 2005. Ucd iirg approach to the textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 53–56.

Niklaus, C.; Bermeitinger, B.; Handschuh, S.; and Freitas, A. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, 170–174. Osaka, Japan: The COLING 2016 Organizing Committee.

Pérez, D., and Alfonseca, E. 2005. Application of the Bleu algorithm for recognising textual entailments. In *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, 9–12. Citeseer.

Raina, R.; Ng, A. Y.; and Manning, C. D. 2005. Robust textual inference via learning and abductive reasoning. In *AAAI*, 1099–1105.

Silva, V. S.; Handschuh, S.; and Freitas, A. 2016. Categorization of semantic roles for dictionary definitions. In *Cognitive Aspects of the Lexicon (CogALex-V), Workshop at COLING 2016*, 176–184.

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37:141–188.

Voorhees, E. M. 2008. Contradictions and justifications: Extensions to the textual entailment task. In *ACL*, 63–71.

Wang, R., and Neumann, G. 2008. An divide-and-conquer strategy for recognizing textual entailment. In *Proceedings of the Text Analysis Conference, Gaithersburg, MD*.

Zhang, K.; Chen, E.; Liu, Q.; Liu, C.; and Lv, G. 2017. A context-enriched neural network method for recognizing lexical entailment. In *AAAI*, 3127–3134.