KnowledgeNet: A Benchmark Dataset for Knowledge Base Population

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza
Diffbot Technologies Corp.
Menlo Park, California
{filipe,matteo,jay,paramita}@diffbot.com

Denilson Barbosa
Department of Computing Science
University of Alberta
Edmonton, Canada
denilson@ualberta.ca

Abstract

KnowledgeNet is a benchmark dataset for the task of automatically populating a knowledge base (Wikidata) with facts expressed in natural language text on the web. KnowledgeNet provides text exhaustively annotated with facts, thus enabling the holistic end-to-end evaluation of knowledge base population systems as a whole, unlike previous benchmarks that are more suitable for the evaluation of individual subcomponents (e.g., entity linking, relation extraction). We discuss five baseline approaches, where the best approach achieves an F1 score of 0.50, significantly outperforming a traditional approach by 79% (0.28). However, our best baseline is far from reaching human performance (0.82), indicating our dataset is challenging. The KnowledgeNet dataset and baselines are available at https://github.com/diffbot/knowledge-net

1 Introduction

Knowledge Bases (KBs) are valuable resources for developing intelligent applications, including search, question answering, data integration, and recommendation systems. High-quality KBs still rely almost exclusively on human-curated structured or semi-structured data. Such a reliance on human curation is a major obstacle to the creation of comprehensive, always-up-to-date KBs.

KB population (KBP) is the task of automatically augmenting a KB with new facts. Traditionally, KBP has been tackled with datasets for individual components to be arranged in a pipeline, typically: (1) entity discovery and linking (Ji et al., 2017; Shen et al., 2015) and (2) relation extraction (Angeli et al., 2015; Zhang et al., 2017). Entity discovery and linking seeks to recognize and disambiguate proper names in text that refer to entities (e.g., people, organizations and locations) by linking them to a reference KB. Relation extraction seeks to detect facts involving two entities (or an entity and a literal, such as a number or date).

KnowledgeNet is a benchmark dataset for populating a KB (Wikidata) with facts expressed in natural language on the web. KnowledgeNet facts are of the form (*subject*; property; *object*), where *subject* and *object* are linked to Wikidata. For instance, the dataset contains text expressing the fact (*Gennaro Basile*[1]; RESIDENCE; *Moravia*[2]), in the passage:

> "Gennaro Basile was an Italian painter, born in Naples but active in the German-speaking countries. *He* settled at Brünn, in *Moravia*, and lived about 1756..."

KnowledgeNet's main goal is to evaluate the overall task of KBP rather than evaluating its subcomponents in separate. We refer to this type of evaluation as *end-to-end*. The dataset supports the end-to-end evaluation of KBP systems by exhaustively annotating all facts in a sentence. For instance, the dataset contains all RESIDENCE facts (two) from the sentence "He settled at Brünn, in Moravia, and lived about 1756". This allows our evaluation to assess precision and recall of RESIDENCE facts extracted from this sentence.

A popular initiative to evaluate KBP is the Text Analysis Conference, or TAC (Getman et al., 2018). TAC evaluations are performed manually and are hard to reproduce for new systems. Unlike TAC, KnowledgeNet employs an automated and reproducible way to evaluate KBP systems at any time, rather than once a year. We hope a faster evaluation cycle will accelerate the rate of improvement for KBP.

In addition to providing an evaluation benchmark, KnowledgeNet's long-term goal is to provide exhaustively annotated training data at large

[1] http://www.wikidata.org/wiki/Q1367602
[2] http://www.wikidata.org/wiki/Q43266

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 749–758,
Hong Kong, China, November 3–7, 2019.

scale. Our goal for the coming years is to annotate *100,000 facts for 100 properties*. To accomplish this goal, we propose a new framework for annotating facts with high accuracy and low effort.

Contributions. Our contributions are as follows. We introduce KnowledgeNet, a benchmark dataset for end-to-end evaluation of KBP systems (Section 3). We propose a new framework for exhaustive yet efficient annotation of facts (Section 3.1). We implement five baseline approaches that build upon state-of-the-art KBP systems (Section 4). Finally, we present an experimental analysis of our baseline approaches, comparing their performance to human performance (Section 5).

2 Related Work

KBP has traditionally been tackled with pipeline systems. For instance, Stanford's TAC 2015 winning system employs the following pipeline: named entity recognition (NER) → entity linking → relation extraction (Angeli et al., 2015). Stanford's latest TAC system continues to use the same pipeline architecture with one additional component: coreference resolution (Chaganty et al., 2017).

The main shortcoming of pipeline systems is error propagation. Mistakes made by components in the beginning of the pipeline are propagated to the final output of the system, negatively affecting the overall precision and recall. For instance, our experiments show that the pipeline employed by Stanford's TAC 2015 winning system can achieve a maximum recall of 0.32 in KnowledgeNet[3].

End-to-end systems (Liu et al., 2018; Miwa and Bansal, 2016) are a promising solution for addressing error propagation. However, a major roadblock for the advancement of this line of research is the lack of benchmark datasets. We hope KnowledgeNet can help support this line of research.

2.1 Datasets

TAC is a series of evaluation workshops organized as several tracks by NIST (Getman et al., 2018). The Cold Start track provides an end-to-end evaluation of KBP systems, while other tracks focus on subtasks (e.g., entity disambiguation and linking). The Cold Start track is the current standard to evaluate KBP systems. To compete in this track, participants have a limited time window to submit the results of their KBP systems. After this window, the systems are evaluated by pooling facts extracted by the contestants. Despite its effectiveness for running a contest, this methodology has been shown to be biased against new systems, which are not part of the pooling (Chaganty et al., 2017). TAC also manually evaluates a system's "justification", a span of text provided as evidence for a fact. A correct fact with an incorrect justification is considered invalid. Therefore, reproducing TAC's evaluation for new systems is challenging.

We propose KnowledgeNet as an automated and reproducible alternative to TAC's evaluation. Before creating KnowledgeNet, we considered using one of the datasets presented in Table 1. We compare these datasets according to five criteria that we consider desirable for a KBP benchmark dataset:

- **Human annotation**: the dataset should be annotated by (multiple) humans to support accurate evaluation.
- **Exhaustive annotation:** for each property, the dataset should exhaustively enumerate all facts of that property that are expressed in the text. Exhaustive annotation allows measuring true precision and recall for a system.
- **Text annotation:** the dataset should contain text spans for entities involved in a fact. This allows evaluating whether the text expresses an extracted fact (as alternative to TAC's manual justification assessments).
- **Links to reference KB:** the dataset should contain a link to the reference KB for entities involved in every fact (or indicate that such an entity doesn't exist in the KB). This allows the evaluation to confirm that the reference KB is being correctly populated.
- **Cross-sentence facts:** the dataset should contain facts involving entities whose *names* never appear in the same sentence. This is because a significant portion of facts expressed in text require coreference resolution of entity mentions spanning multiple sentences.

ACE 2005[4] is a popular dataset for end-to-end relation extraction systems (Li and Ji, 2014;

[3]Maximum recall is the recall of candidate facts, which are used as input to the last component of the pipeline (relation extraction).

[4]https://catalog.ldc.upenn.edu/LDC2006T06

Dataset	**KnowledgeNet**	ACE	TAC	TACRED	FewRel	DocRED	GoogleRE	T-REx
Human annotation	yes	yes	yes	yes	yes	yes	yes	no
Exhaustive annotation	yes	yes	no	no	no	no	no	no
Exhaus. anno. sentences	9,000	11,000	N/A	N/A	N/A	N/A	N/A	N/A
Text span annotation	yes	yes	no	yes	yes	yes	no	yes
Links to reference KB	yes	yes	yes	no	yes	no	yes	yes
Cross-sentence facts	yes	yes	yes	no	no	yes	yes	yes
Annotated facts	13,000	8,000	84,000	22,000	56,000	56,000	40,000	11M
Properties	15	18	41	41	100	96	5	353
New KB facts annotated	77%	100%	100%	100%	0%	100%	0%	0%

Table 1: A dataset comparison according to our criteria for a desirable KBP benchmark dataset. "Exhaus. anno. sentences" shows the number of exhaustively annotated sentences. "New KB facts annotated" shows the percentage of annotated facts that can be found in the reference KB. Most datasets contain only facts with no links to a reference KB (100% new facts) or contain only facts that exist in the KB (0% new facts).

Miwa and Bansal, 2016). According to our criteria, ACE might seem the most complete benchmark dataset for end-to-end evaluation of KBP. It exhaustively annotates every sentence of 599 documents with mentions, coreference chains and facts for 18 properties. Also, ACE has been independently extended with links to Wikipedia (Bentivogli et al., 2010). However, a closer look at ACE's annotations reveals that most of them are ill-suited for general-purpose KBs. These annotations include facts about broad properties (e.g., part-whole, physical location) or mentions that do not refer to named entities (e.g., "two Moroccan men", "women on the verge of fainting", "African immigrants who came ashore Thursday"). Perhaps not coincidently, we are unaware of any work using it for the purpose of evaluating a KBP system.

Our annotation framework (Section 3.1) is inspired by ACE's framework but tailored towards KBP. First, we only annotate mentions that refer to named entities. Second, while our annotation is exhaustive, we focus on annotating sentences rather than documents, eliminating the need to annotate every fact described in the entire document. Such a requirement creates a significant imbalance in the number of annotations per property. For instance, the most popular property from ACE has 1,400 annotated facts, while the majority of properties from ACE have less than 300 annotated facts. This might explain why most relation extraction evaluations use only 6 properties from ACE.

Annotating every sentence with facts for all properties is also detrimental to the incremental nature of KnowledgeNet. Adding one property to the dataset would require an annotator to re-annotate every sentence in the dataset. In contrast, our framework selects a limited set of sentences to be annotated for a particular property. The remaining sentences are ignored during annotation and evaluation. As a consequence, our annotation framework allows incremental annotation of new properties and is better suited for the goal of annotating 100,000 facts for 100 properties.

Datasets employing non-exhaustive annotation. Recent datasets like T-REx automatically annotate facts in text as a way to produce training data cheaply. This is performed by aligning facts in the KB to sentences referring to them (Elsahar et al., 2018). Other datasets go further and use human annotators to label every alignment as correct or incorrect. These semi-supervised datasets include TACRED (Zhang et al., 2017), GoogleRE[5], FewRel (Han et al., 2018) and DocRED (Yao et al., 2019). Annotations created in this way are useful for training KBP systems. However, they do not provide an exhaustive annotation of facts, which is needed for end-to-end evaluation of KBP. For instance, Zhang et al. (2017) train their KBP system with TACRED, but rely on TAC to evaluate the system.

3 KnowledgeNet Dataset

This section discusses the first release of KnowledgeNet and our annotation framework. The documents in this first release are either DBpedia abstracts (i.e., first paragraphs of a Wikipedia page) or short biographical texts about a person or organization from the web. These web texts were collected using the Diffbot Knowledge Graph[6].

Table 2 presents the number of annotated facts for each property. We chose 9,073 sentences

[5] https://code.google.com/archive/p/relation-extraction-corpus/downloads

[6] https://www.diffbot.com/

Property	Facts	Sent.	Relevant
DATE_OF_BIRTH (PER–DATE)	761	731	468
DATE_OF_DEATH (PER–DATE)	664	512	347
RESIDENCE (PER–LOC)	1,456	796	387
BIRTHPLACE (PER–LOC)	1137	936	407
NATIONALITY (PER–LOC)	639	801	396
EMPLOYEE_OF (PER–ORG)	1,625	650	543
EDUCATED_AT (PER–ORG)	951	463	335
POLITICAL_AFF. (PER–ORG)	635	537	318
CHILD_OF (PER–PER)	888	471	296
SPOUSE (PER–PER)	1,338	504	298
DATE_FOUNDED (ORG–DATE)	500	543	315
HEADQUARTERS (ORG–LOC)	880	564	296
SUBSIDIARY_OF (ORG–ORG)	544	481	299
FOUNDED_BY (ORG–PER)	764	558	346
CEO (ORG–PER)	643	526	350
Total	13,425	9,073	5,423

Table 2: KnowledgeNet properties and their number of annotated facts and sentences. "Relevant" indicates the number of relevant sentences (i.e., those with one or more annotated facts). Subjects and objects belong to one of the following types: person, organization, location and date.

from 4,991 documents to be exhaustively annotated with facts about a particular property of interest. Because our annotation is exhaustive, negative examples of facts can be automatically generated. In total, KnowledgeNet comprises 13,425 facts from 15 properties.

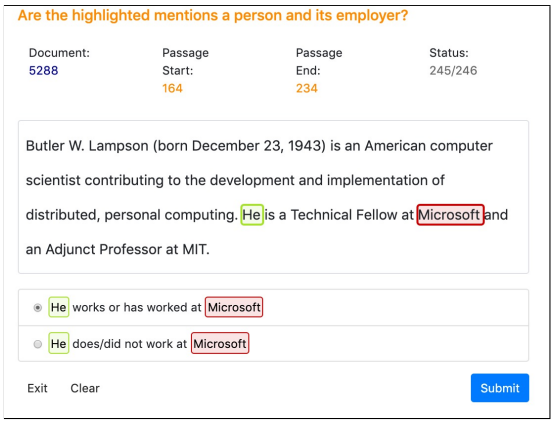
Holdout test set. We split the documents into five folds in a round-robin manner, keeping the fifth fold (20% of the dataset) as the test set. To preserve the integrity of the results, we will release the test set without annotations and will provide a service through which others can evaluate their KBP systems. In our experiments, we used folds 1-3 for training and fold 4 for development and validation, including hyperparameter tuning.

3.1 Dataset Annotation

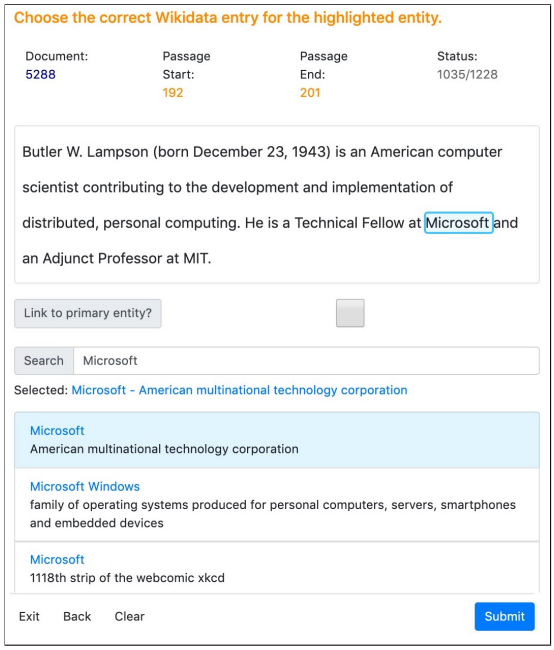
The dataset has been generated by multiple annotators using a new multi-step framework. We conjecture our framework can help annotators produce higher quality annotations by allowing them to focus on one small, more specific task at a time. The annotation consists of four different steps: (1) fetch sentences, (2) detect mentions, (3) classify facts and (4) link entities.

Step 1: Fetch sentences. We employ two methods of choosing a sentence for annotation. The first method leverages T-REx's automatic alignments (Elsahar et al., 2018) to find sentences that are likely to describe facts from Wikidata. The

(a) Interface to detect mentions of an entity type.



(b) Interface to classify facts.



(c) Interface to link a mention to a Wikidata entity.

Figure 1: Interface for Steps 2-4 of our framework. Step 1 fetches sentences to be exhaustively annotated for one property. The remaining steps guide annotators to detect entity mentions, facts and links in each sentence.

second method chooses sentences that contain a keyword that might indicate the presence of a fact for a property (e.g., *"born"* for DATE_OF_BIRTH). We have chosen these keywords by leveraging Wikidata's "also known as" values for properties as well as WordNet synonyms. By using these keywords, we prevent the dataset to be exclusively annotated with facts that are known in Wikidata. In fact, only 23% of facts annotated in this release are present in Wikidata.

For each fetched sentence, an annotator decides whether the sentence is *relevant* for the property of interest (i.e., whether this sentence describes one or more facts for this property). Relevant sentences go through steps 2 through 4; while irrelevant sentences are kept to be used for detecting incorrectly extracted facts (i.e., false positives).

It is worth noting that this step might not fetch some relevant sentences. Our framework does not require all relevant sentences to be annotated and does not penalize systems for extracting facts from sentences that were not annotated.

Step 2: Detect mentions. In this step, we ask annotators to highlight entity names (Figure 1a). We consider only entities whose type is relevant to the property being annotated. For instance, an annotator will only be asked to highlight names of people and organizations when annotating the property FOUNDED_BY. Pronouns are automatically annotated with a gazetteer. To decrease the likelihood of missing a mention, we consider the *union* of mentions highlighted by two annotators for the following step.

Step 3: Classify facts. We ask annotators to classify a candidate fact (i.e., a pair of mentions) in a sentence as a positive or negative example for a property (Figure 1b). Each candidate fact is annotated by at least two annotators. A third annotator breaks the tie when the first two annotators disagree.

We follow ACE's *reasonable reader rule*, which states that a fact should only be annotated when there is no reasonable interpretation of the sentence in which the fact does not hold. In other words, annotators are asked to only annotate facts that are either explicitly stated in the sentence or inferred with absolute certainty from the sentence alone (i.e., without using external world knowledge).

Step 4: Link entities. Finally, we ask annotators to link every mention involved in a fact to a single Wikidata entity. In this step, annotators can read the entire document and resolve mentions (e.g., pronouns) that refer to names in other sentences. Every mention is annotated by at least two annotators. When there is disagreement, we ask other annotators to join in the process until consensus is reached. In total, excluding the properties having literal objects (Table 2) we can assign a link to both subject and object for 52% of the facts.

Inter-annotator agreement. A total of five annotators have contributed to KnowledgeNet so far. In Step 3, the initial two annotators have annotated 33,165 candidate facts with 96% agreement. They disagreed on 1,495 candidate facts, where 599 have been deemed positive by a third annotator. In Step 4, the initial two annotators have annotated 13,453 mentions with agreement of 93%. The remaining 7% of mentions were resolved with additional annotators.

Timing. On average, annotating a sentence for one property takes 3.9 minutes. This total time includes two annotators (plus additional annotators for tiebreaking). It also includes inspecting sentences that express no facts and therefore do not go through steps 2-4 (but are included in the dataset and are helpful for assessing false positives). The most expensive step is Step 3 (40% of the total time), followed by Step 4 (28%), Step 2 (22%) and Step 1 (10%).

3.2 Limitations

Our first release is comparable to other benchmarks in size (e.g., ACE 2005), but it is perhaps insufficient to train data-hungry models. This is by design. Most organizations do not have the resources to produce tens of thousands of examples for each property of interest. As we expand the number of properties to achieve our goal of annotating 100,000 facts, we expect to keep the number of facts per property to around a thousand. In this way, we hope to promote approaches that can learn from multiple properties, requiring less annotations per property. We also hope to promote approaches using KnowledgeNet together with semi-supervised or unsupervised datasets for training.

Another limitation of our first release is the focus on individual sentences. Currently, our framework can only annotate a fact when the subject and

the object are explicitly mentioned by a name or pronoun in a sentence. Others have reported that the majority of facts fall into this category. For example, the authors of DocRED report that 41% of facts require reasoning over multiple sentences in a document (Yao et al., 2019). This indicates that a fact's subject and object are mentioned by their full name in a single sentence 59% of the time. The percentage of facts that can be annotated in KnowledgeNet is significantly higher than 59%. This is because our framework can also annotate facts that require resolving (partial) names and pronouns referring to full names in other sentences. These facts are particularly common in our document collection.

4 Baseline Approaches

This section presents five baseline approaches for KBP. We evaluate these approaches and compare their performance relative to human annotators in Section 5.

Figure 2 illustrates the architecture shared by our five baseline approaches. We start by splitting a given document into sentences. For each sentence, we detect entity mentions using a named entity recognizer (NER) and a gazetteer for pronoun mentions and their type (e.g., person, organizations, location). We also detect coreference chains, that is, groups of mentions within a document that refer to the same entity. Figure 2 illustrates how coreference chains help disambiguate pronouns and partial names by clustering them together with the full name of an entity. Finally, we link these coreference chains to Wikidata entities.

Next, we produce candidate facts by considering pair of mentions from the same sentence, as illustrated in Figure 2. The relation extraction component makes the final decision on whether a candidate fact is expressed by the text.

4.1 Relation Extraction

Figure 2 illustrates our relation extraction model. This model follow the literature by using a Bi-LSTM network (Miwa and Bansal, 2016; Xu et al., 2015; Zhou et al., 2016; Zhang et al., 2017), which is effective in capturing long-distance dependencies between words. We train a single multi-task model for all properties using both positive examples (i.e., annotated facts) and automatically generated negative examples.

The model outputs two values for each property. The first value represents the likelihood of the subject and object *mentions* (i.e., text spans) to be correct, while the second value represents the likelihood of the subject and object *links* to be correct. We learn individual thresholds for each value and property. When both values are above the threshold, the system outputs the fact with links. When the first value is above the threshold and the second value is below the threshold, we output the fact without links.

Features. Figure 3 illustrates features encoding syntactic and positional information, which are concatenated to the embedding of every word.

1. **Enriched NER:** NER label for names (using a NER system) and pronouns (using gazetteers for each type).
2. **Mention distance:** distance between each word and the subject and object mention, following Zhang et al. (2017).
3. **Shortest dependency path (SDP) length:** number of edges in the SDP between the word and the subject and object.
4. **SDP distance:** number of edges separating the word to the closest word in the SDP between the subject and object.
5. **Coreference confidence:** confidence score of the coreference resolution system that a word refers to the subject and object.
6. **Coreference distance:** distance to the closest mention in the coreference chain of the subject and object.
7. **Coreference SDP length:** number of edges in the SDP between the word and the closest mention in the subject and object chain.
8. **Coreference SDP distance:** number of edges separating the word to the closest word in the SDP between the subject and object coreference chains.
9. **KB entity types:** entity types for the subject and object from Wikidata.
10. **KB properties:** the property p where (*subject*; p; *object*) exists in Wikidata (when both the subject and object have links).

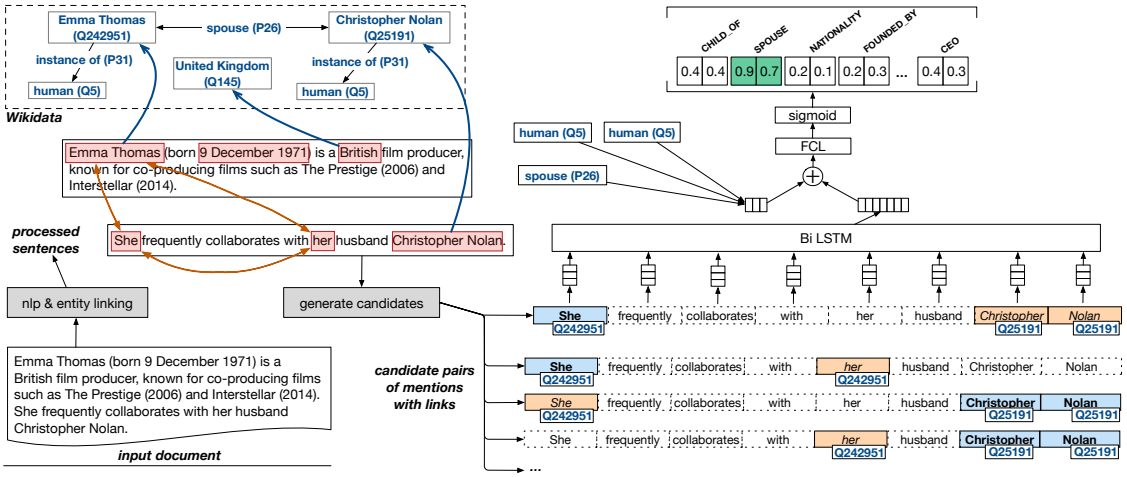


Figure 2: The architecture of our baseline approaches, illustrated with an example. Red arrows and boxes represent coreference chains and blue arrows represent links to Wikidata. The subject and object of candidate facts are highlighted in bold (blue) and italics (orange), respectively.

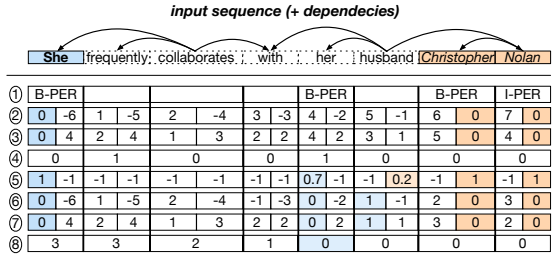


Figure 3: Features representing the relationships between the words. Significant relationships with the subject and object are highlighted in blue and orange, respectively.

Features 9 and 10 are generated by querying Wikidata and are relative to a single entity pair. We concatenate those features to the Bi-LSTM output, as illustrated in Figure 2.

4.2 Baseline Approaches

We propose five baselines obtained by improving the candidate generation and relation extraction components.

Baseline 1. Our first baseline is a standard pipeline approach inspired by the TAC 2015 Cold Start winning system (Angeli et al., 2015). It generates candidate mentions by using NER and the pronoun gazetteers. For mentions of the correct type (e.g., person for the property SPOUSE), the system then links these mentions using an entity linking system. The relation extraction component uses features 1-4.

Baseline 2. Our second baseline adds coreference resolution. This baseline is inspired by Stanford's TAC 2017 system (Chaganty et al., 2017). We leverage coreference chains to both increase the number of candidate mentions linked to KB entities (e.g., pronouns) as well as to introduce additional features. This model uses features 1-8.

Baseline 3. Our third baseline adds features 9 and 10 to the relation extraction model. These features leverage Wikidata information for the linked entities, such as entity types and known facts.

Baseline 4. Our fourth baseline seeks to decrease error propagation by allowing more candidate facts to be evaluated by the relation extraction component. This is done in two ways. First, Baseline 4 uses all mentions regardless of their NER type when creating candidate facts. Second, this baseline adds a candidate link to mentions that had no candidate link in Baseline 1-3 (due to incorrect coreference chains). This is done by choosing a link outside of the mention's coreference chain that maximizes a combination of entity linking score and coreference resolution score.

Baseline 5. Our final baseline seeks to improve the relation extraction component by employing BERT's pre-trained representations (Devlin et al., 2018) in addition to all other features. To produce a contextual representation for every word, we learn a linear weighted combination of BERT's 12 layers, following Peters et al. (2019).

4.3 Implementation

All our baseline systems follow the same architecture (Figure 2). We use spaCy[7] for the NLP pipeline (sentence splitting, tokenization, POS tagging, dependency parsing, NER), Hugging Face's coreference resolution system[8], and the Diffbot Entity Linker[9] for entity linking.

For relation extraction we implement a standard BiLSTM network with two 500-dimensional hidden layers. We use spaCy pre-trained word embeddings (size 300) concatenated with additional features illustrated in Figure 3. The output of the BiLSTM network is concatenated with features from Wikidata (Features 9-10).

We train all the networks using mini-batches of 128 examples and Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. We use the fourth fold of the dataset as validation set, selecting the model that minimize the loss function value. The same validation set is used to find thresholds for the output values that maximize the F1 score for each property.

5 Experiments

Table 3 presents the performance of our baseline systems compared to the human performance. We report precision (P), recall (R) and F-score ($F1$):

$$P = \frac{\text{correctly extracted facts}}{\text{extracted facts}},$$

$$R = \frac{\text{correctly extracted facts}}{\text{annotated facts}},$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R}.$$

We evaluate our baseline systems from two perspectives. The *text evaluation* deems an extracted fact correct when the text spans of the subject and object overlap with the text spans of a ground truth fact. The *link evaluation* deems an extracted fact correct when the links of the subject and object match the links of a ground truth fact. In the link evaluation, we consider only facts where both the subject and object links are present.

Human performance. To measure the human performance on the end-to-end KBP task, one of our annotators was asked to enumerate all facts described in a sample of the test sentences. We report the performance of our annotator in Table 3.

[7] https://spacy.io/
[8] https://huggingface.co/coref/
[9] https://diffbot.com/

System	Text evaluation			Link evaluation		
	P	R	F1	P	R	F1
Baseline 1	0.44	0.64	0.52	0.31	0.26	0.28
Baseline 2	0.49	0.64	0.55	0.37	0.32	0.34
Baseline 3	0.47	0.66	0.55	0.35	0.37	0.36
Baseline 4	0.60	0.65	0.62	0.51	**0.48**	0.49
Baseline 5	**0.68**	**0.70**	**0.69**	**0.53**	**0.48**	**0.50**
Human	0.88	0.88	0.88	0.81	0.84	0.82

Table 3: The performance of our baseline approaches is well below human performance.

A closer look at the annotator's mistakes shows that 32% of the mistakes are due to incorrect annotations in KnowledgeNet (i.e., the annotator is actually correct). The remaining mistakes (68%) are mostly due to the annotator entering an incorrect fact (30%) or missing a link on a correct fact (18%). These results show that our annotation framework produces significantly better annotations than individual annotators working without our framework.

Baseline performance. Table 3 presents the performance of our baselines. Our best baseline (Baseline 5) significantly outperforms the standard pipeline approach (Baseline 1) in both the text and link evaluation. However, the performance of Baseline 5 is well below the human performance. The most impactful improvements over Baseline 1 are due to (a) incorporating coreference when choosing candidate links for pronouns in Baseline 2; (b) allowing more candidate facts and links to be classified by the relation extraction component in Baseline 4; and (c) incorporating BERT's pre-trained model in Baseline 5.

Table 4 shows the "maximum recall" for each baseline (i.e., recall of candidate facts used as input for the relation extraction component). These results indicate that error propagation significantly limits recall. Our best baseline shows higher maximum recall due to coreference resolution (introduced in Baseline 2) and removing the filtering of candidate facts based on NER types (introduced in Baseline 4). The low maximum recall for link evaluation is mainly due to incorrect candidate links, which can only be omitted (but not fixed) in our baselines.

6 Conclusion

We introduce KnowledgeNet, an end-to-end benchmark dataset for populating Wikidata with facts expressed in natural language text on the

System	Text evaluation	Link evaluation
	Maximum Recall	Maximum Recall
Baseline 1	0.80	0.33
Baseline 2	0.80	0.37
Baseline 3	0.80	0.37
Baseline 4	0.90	0.59
Baseline 5	0.90	0.59

Table 4: The relation extraction component's recall is limited by error propagation. Maximum recall is the recall of the candidate facts used as input for the relation extraction component on the dev set.

web. We build KnowledgeNet using a new multi-step framework that helps human annotators to produce high-quality annotations efficiently. We also introduce five baseline systems and evaluate their performance. Our best baseline outperforms a traditional pipeline approach by 79% (F1 score of 0.50 vs. 0.28). Human performance is significantly higher (0.82), indicating that KnowledgeNet can support further research to close this gap.

Our experiments show that the traditional pipeline approach for KB population is notably limited by error propagation. Performance gains achieved by our best baseline are mainly due to more candidates being passed along to the final pipeline component (relation extraction), allowing this component to fix errors made by previous components. A closer inspection reveals that even our best baseline is fairly limited by error propagation and can only achieve a maximum recall of 0.59. These results indicate that end-to-end models might be a promising alternative to the traditional pipeline approach.

Acknowledgments

We would like to thank Veronica Romualdez and Geraldine Fajardo for their diligent annotation work. We would also like to thank Mike Tung, Zhaochen Guo, Sameer Singh and the anonymous reviewers for their helpful comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Diffbot.

References

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Tejasvi Chaganty, Jason Bolton, Melvin Jose Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D. Manning. 2015. Bootstrapped self training for knowledge base population. In *TAC*. NIST.

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. 2010. Extending English ACE 2005 corpus annotation with ground-truth links to Wikipedia. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 19–27, Beijing, China. Coling 2010 Organizing Committee.

Arun Chaganty, Ashwin Paranjape, Percy Liang, and Christopher D. Manning. 2017. Importance sampling for unbiased on-demand evaluation of knowledge base population. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1038–1048, Copenhagen, Denmark. Association for Computational Linguistics.

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

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song, and Jennifer Tracey. 2018. Laying the Groundwork for Knowledge Base Population: Nine Years of Linguistic Resources for TAC KBP. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4803–4809.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, and Cash Costello. 2017. Overview of TAC-KBP2017 13 languages entity discovery and linking. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Yue Liu, Tongtao Zhang, Zhicheng Liang, Heng Ji, and Deborah L. McGuinness. 2018. Seq2rdf: An end-to-end application for deriving triples from natural language text. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018.*

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. pages 1105–1116.

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.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019.*, pages 7–14.

Pouya Pezeshkpour, Liyan Chen, and Sameer Singh. 2018. Embedding multimodal relational data for knowledge base completion. *CoRR*, abs/1809.01341.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743.

Peng Xu and Denilson Barbosa. 2019. Connecting language and knowledge with heterogeneous representations for neural relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, Minnesota, USA, June 2-7, 2019, Volume 2 (Short Papers)*, page 4.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 764–777.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

A Beyond Binary Relationships

While it would be convenient to express all facts as (*subject*; property; *object*) triples, this is not always possible. Many facts require further annotations to be sufficiently and accurately expressed in the KB. Take for instance (*United States*; head_of_government; *Barack Obama*), which only holds true in the past.

Qualifiers allow facts to be expanded or contextualized beyond what can be expressed with binary relationships. More specifically, qualifiers can be used to constrain the validity of a fact in time or space, e.g., (*employment fact*; end_time; *2017*); represent n-ary relationships, e.g., (*casting fact*; character_role; *Tony Stark*); and track provenance.

This release contains 4,518 facts annotated with three *temporal qualifiers*: IS_CURRENT, START_TIME and END_TIME. We use one of our baseline system to obtain facts to be annotated with qualifiers, along with the the sentence where each fact was found. Given a fact and a sentence, human annotators must decide the value of a qualifier (*true* or *false* for IS_CURRENT or a time expression for START_TIME, END_TIME). A third option *unclear* can be chosen in the case of uncertainty. To be included in the dataset, each fact must be annotated by two annotators in agreement. While preliminary experiments show promising results for qualifier extraction, they are out-of-scope of this work.