Revisiting the Task of Scoring Open IE Relations

William Léchelle, Philippe Langlais
RALI - University of Montreal
{lechellw, felipe} @ iro.umontreal.ca

Abstract

Knowledge Base Completion infers missing facts from existing ones in knowledge bases. As recent Open Information Extraction systems allow us to extract ever larger (yet incomplete) open-domain Knowledge Bases from text, we seek to probabilistically extend the limited coverage we get from existing facts, to arbitrary queries about plausible information. We propose a simple baseline, based on language modeling and trained with off-the-shelf programs, which gives competitive results in the previously defined protocol for this task, and provides an independent source of signal to judge arbitrary fact plausibility. We reexamine this protocol, measure the (large) impact of the negative example generation procedure, which we find to run contrary to the belief put forward in previous work. We conduct a small manual evaluation, giving insights into the rudimentary automatic evaluation protocol, and analyse the shortcomings of our model.

Keywords: Open Information Extraction, Evaluation, Knowledge Base Completion

1. Introduction

Much recent effort has been put into building Knowledge Bases (KBs), either manually curated (Freebase (Bollacker et al., 2008), Cyc (Lenat, 1995)) or automatically produced (YAGO (Suchanek et al., 2007), Knowledge Vault (Dong et al., 2014)), ranging from logically consistent linked-data in OWL (SUMO (Pease et al., 2002)) to little-structured sets of textual relations extracted from text (NELL (Mitchell et al., 2015)) with Open IE systems (Reverb, Ollie (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013), Stanford Open IE (Angeli et al., 2015), CSD-IE (Bast and Haussmann, 2013)). However large they may be, typical KBs are largely incomplete, and many relevant facts are missing (West et al., 2014).

Because an exhaustive coverage of the information that ought to be part of the KB is a very desirable feature, KB completion (inference of missing facts from known ones) is a rapidly growing field (West et al., 2014; Nickel et al., 2015; Wang et al., 2015; Toutanova et al., 2016).

In the context of Open Information Extraction (OIE), our aim is to assign a score to an arbitrary unseen query fact[1], judging its *plausibility* as a member of the KB. This task is important for three reasons : first, it extends the coverage of the existing KB probabilistically to any query, greatly improving upon the closed-world assumption that facts not known to be true are false. Second, as the extraction of information in the open domain is a relatively noisy process, a confidence score helps detecting extraction errors, and makes for higher-quality automatically generated KBs. Last, adjusting the confidence threshold of extracted facts allows to tune as desired the trade-off between precision and recall of the extraction process.

The task has attracted little attention since it was introduced in (Angeli and Manning, 2013), most that we know in (Li et al., 2016). The authors propose to assign high KB-membership probability to facts that have *support facts* existing in the KB, which are similar to them (based on common phrases and word similarity).

We propose a new baseline for this task, in the form of a language model. Whereas the method of (Angeli and Manning, 2013) is fairly complicated to implement, and requires indexing the KB in various ways for intermediate computations, a trained language model is very compact, straightforward to implement and train, and fast to process requests at use time. We train the model on automatically extracted facts (including some noise) from the same corpus, i.e. the knowledge base, taken as a list of sentences. We experiment with language model features and show that a linear classifier gives good results at the task of recognising actual extracted facts, perhaps unsurprisingly.

We then go back over the experimental protocol proposed by (Angeli and Manning, 2013) and consider the way negative examples are automatically generated. We find that this procedure has a significant impact on the difficulty of the task. At last, we go back to the goal of improving existing extractions by picking out the noise. Instead of an automatically generated test set, we measure the ability of the models to identify the remaining wrong extractions in the RV-15M high-quality dataset (presented in section 4.1.).

2. Related work

2.1. Knowledge Base Completion

Much work in Knowledge Base Completion (KBC) has been done in recent years (Bordes et al., 2013; Riedel et al., 2013; García-Durán et al., 2015; Feng et al., 2016; Trouillon et al., 2016), on tasks very similar to ours, mostly focusing on Freebase, and other such large manually curated KBs (WordNet, NCI-PID (Schaefer et al., 2008), etc.).

The major difference between our approach and most KBC work is the predefined schema of the KB. The arguments of the relations curated in Freebase are mostly named entities, and the relations to be gathered were defined when building the KB. FB15k, a popular Freebase dataset, covers 1345 predicates, though only 401 have more than 100 occurences (Yang et al., 2014). NELL captures about 150 relations, and WordNet about 20. By contrast OIE seeks to extract all the relations expressed in text, resulting in hundreds of

[1] A "fact" is a relation phrase linking two or more argument phrases. The arguments need not be named entities.

thousands of relation predicates (even though many are synonyms). The RV-15M dataset used in this work has 660k distinct relation strings.
(Toutanova et al., 2015) embed surface textual patterns in the same vector space as the KB relations, which is similar to the implicit embedding of all predicates in the same vector space as we do. Yet their work only predicts relations based on the 237 predicates of the FB15k-237 dataset, whereas we predict confidence scores for all relations, including predicates never seen during training.

2.2. Angeli and Manning (2013)

In this work we reexplore in depth the task set up in (Angeli and Manning, 2013), and it is their work that is most similar to ours. They seek to probabilistically extend a KB to arbitrarily any query fact, in the sense that any candidate fact has a KB-membership probability (or confidence score). Indeed, this is a sensible way of considering a knowledge base system that must perform inferences.
To this aim, they compare a query fact to a set of candidate support facts from the KB. The candidate facts need to share 2 phrases (arguments or relation) with the query fact, and is allowed to differ by the third part. The query fact has a high KB-membership score if it is similar enough to its closest support facts (that is, the differing part is similar enough). The algorithm is as follows:

- Gather candidate facts that can support the query fact: candidates must share two of (argument 1, relation, argument 2) with the query, these phrases having at least the same head word. Stricter criteria are used as long as there is a sufficient number of candidates.
- Compute the distances between the query fact and each of its supporting facts, using 11 distance metrics, based on both distributional similarity and the WordNet thesaurus - cosine, Jaccard, etc.
- The highest similarities are used as weights in a linear classifier, whose job is to agreggate the similarity values across candidates and distance metrics.

2.3. OIE Systems Confidence

Contrarily to most OIE systems in which the confidence score is often an afterthought, Fader et al. (2011) went to great lengths to develop Reverb's confidence function (see their Section 4.2). They manually labeled the extractions from 1000 Web sentences as correct or incorrect, and trained a classifier using features about the *original sentence* to assign a confidence score to the extraction process. The confidence function of Ollie is based on the frequency of the syntactic pattern that was used to extract a given fact. ClausIE simply returns the confidence score of the underlying dependency parser, as its rules rely directly on it.
In contrast to our work, the confidence scores produced by OIE systems rely on the original sentence from which they were extracted, and on how well the extraction procedures could handle the input sentences. Our goal is to judge the quality of query facts as they stand, regardless of the sentences they come from. For instance, with the Chomsky sentence *Colorless green ideas sleep furiously* as input, past OIE evaluation would consider (*ideas, sleep, furiously*) to be a correct extractions, whereas the goal of our system is to reject any "fact" coming from that sentence on the grounds that they do not make sense.

3. Revisiting the Task Setup

Our end goal is to improve the OIE process by pruning out the erroneous or empty facts produced. We frame this as a classification problem, seeking to distinguish correct useful facts from ill-constructed or void statements.

3.1. Task Protocol

A KB is constructed by running an open information extractor over a textual corpus. Even though there is some noise, extracted facts are assumed to be correct, and samples of them are set aside from the KB to constitute positive examples of unseen facts for the classification task.
For negative examples, artificial facts are constructed by replacing one part of a genuine extraction with that of another. Let (a_1, r, a_2) and (a'_1, r', a'_2) be two genuine facts from the KB, then one negative example is picked between (a'_1, r, a_2), (a_1, r', a_2) and (a_1, r, a'_2). We will show in Section 4.3. that this choice is a very significant parameter of the experimental setup.
Classifiers are trained to discriminate the positive from the negative examples. The performance metric is the classification accuracy.

3.2. Approaches

ArgSim is a weak baseline for the task, measuring the cosine similarity between a_1 and a_2 in a fact. Arguments (effectively bag-of-words) are represented by the average of their individual words' embeddings. This performs well on ConceptNet, per (Li et al., 2016), but captures little information on OIE facts. This is both reported by Angeli and Manning (2013) and replicated in our experiments.
The full algorithm of (Angeli and Manning, 2013), presented in Section 2.2., is denoted **AM-system** in Table 1. We reimplemented the **count** and **cos** methods used in their evaluation, which are both simplified versions of their approach (the former coarse, and the latter close in principle to the full-fledged scoring function).
As has been noted by (Stanovsky et al., 2015), OIE output can be used as training material for other tasks such as text comprehension, word similarity and analogy. This is because OIE produces a distinctive intermediate representation of the sentence, from which complementary features (to that of dependency parse or lexical representations) can be extracted.
Moreover, in the confidence scoring function of Reverb, several features capture how completely the extraction covers the sentence's tokens. In short, the most typical correct extractions look like short declarative sentences, like (*Hudson*, *was born in*, *Hampstead*), or (*Hampstead*, *is a suburb of*, *London*). Then, it seems natural to train a language model on confidently extracted facts, and to expect from correct unseen extractions that they fit well and cause low perplexity. This implements the assumption that an unseen fact is plausible iff it resembles a short and natural-sounding sentence.

The most basic implementation of this idea (**LM-basic**) is to straightforwardly use the probability of concatenated $(a_1.r.a_2)$ as its score. Next, we notice that given the way negative facts are constructed, all three argument and relation spans are probable as they stand (since they come from genuine extractions). What makes the fact incorrect is that the parts do not fit together. Therefore, we use as a score the (log) probability of the whole fact minus that of each individual part. We call this model **LM-junctions**. With $\{a_1, r, a_2\}$ the fact expressing that the relation r holds between the two arguments a_1 and a_2, and p the language model probability function : $\text{score}(\{a_1, r, a_2\}) = \log p(a_1.r.a_2) - \log p(a_1) - \log p(r) - \log p(a_2)$.

We trained the language models with KenLM (Heafield, 2011), using the knowledge base itself as a corpus, each fact being considered as a sentence. We trained 5-gram models (the default), doing as little parameter tuning as possible.

Further, we train a linear classifier based on linguistic modeling-based features (**LM-SVM**). We implemented the SVM with scikit-learn (Pedregosa et al., 2011)[2]. It uses 20 features such as the individual log-probabilities of the various parts, the log-probabilities of various bigrams and trigrams focused on the argument-relation junctions, and arithmetic operations over those values.

The classifier is trained on 10k genuine extractions as positive examples, and 10k artificially constructed ones as negative examples, counts picked to be the same as those in (Angeli and Manning, 2013).

4. Experiments

4.1. Dataset

We used Reverb-15M, a shared[3] dataset of high-quality binary assertions extracted by Reverb on the ClueWeb09 corpus. In order to obtain a high-precision dataset, its authors filtered the extractions by Reverb's confidence function (with a 0.9 threshold), stopwords, and frequency, along with certain syntactic criteria.[4] We used the normalized (lemmatized) version of the tuples. Taken as a text corpus, the RV-15M dataset is 98M tokens long.

Angeli and Manning (2013) used a similar set of extractions : the authors ran Reverb over ClueWeb09 themselves, filtered out extractions scoring under 0.5 per Reverb's confidence function, and retained the first billion extractions, which results in a KB of 500 million unique facts. Their dataset is thus larger, and noisier, than the one we used.

4.2. Classification

Table 1 shows the performance of our approach, along with the methods **count** and **cos** of (Angeli and Manning, 2013). The most elementary language modeling idea demonstrably captures some useful signal for the task. By focusing on the probability of the argument-relation junctions, the language model improves to near state-of-the-art performance.

	Reverb-15M (AUPRC)	Reverb-500M†
Random	50.0 (0.500)	50.0
ArgSim	53.0 (0.283)	52.6
AM-count	61.6 (0.399)	52.3
AM-cos	64.2 (0.462)	70.6
AM-system		74.2
LM-basic	65.4 (0.471)	
LM-junctions	73.6 (0.589)	
LM-SVM	76.3 (0.628)	

Table 1: Classification accuracy of the scoring methods on Reverb-15M, evaluated on the automatically-generated test set. Area under the precision-recall curve is indicated in parenthesis. † Results on Reverb-500M (a similar, larger and noisier dataset) published in (Angeli and Manning 2013), are reproduced for comparison.

By training a linear classifier on top of the language modeling features, we can gain 3 additional points in precision, surpassing previous state-of-art performance.

Examples of correct tuples scored highly by our model are (*Austin Airways, was an airline based in, Canada*) and (*Children, are welcome in, the Curriculum Lab*).

Reversely, (*A knight, can turn into, a serious problem*) and (*Hand, be in, The Los Angeles Times*) are examples of incorrect tuples scored highly. The first argument of the former was originally '*A bad sunburn*' and the relation of the latter was '*has also written for*'. Both facts are plausible and show the limits of the automatic evaluation procedure (*be in* is accidentaly close in meaning to *has also written for*, for a journalist and a newspaper).

Correct tuples scored poorly include (*A bounty Hunter, sent to kill ; Gust*) and (*Casey, also works on, pair-NIC*). Our model finds high perplexity in infrequent proper nouns. A Named Entity Recognition module would certainly improve the model, so that *<PERSON> also works on <MISC>* is seen at training time (or training data featuring the entities of interest if the KB focuses on a certain domain). Bad tuples correctly identified include (*81.45% of total wagers, can also avail the services of, bettors*) and (*A trellis, can also refer to, permission of the artist a structure*), in which relation and second argument formerly were '*returned to*' and '*a structure*' respectively.

4.3. Impact of Negative Examples Sampling Method

One important task parameter is the way negative examples are constructed. With (a_1, r, a_2) and (a_1', r', a_2') two genuine extractions, then one of (a_1', r, a_2), (a_1, r, a_2'), or (a_1, r', a_2) will be used as a negative example. We examine the impact of choosing one of the former two (changing an argument) versus picking the latter (changing the relation[5]). In Figure 1, we vary the fraction of relation changes over argument changes (a "knob" of the task setup), and measure the ensuing precision of systems.

[2]We shortly experimented with Logistic Regression and Random Forest classifiers, giving slightly inferior results. We also tried several available SVM kernels, the default RBF kernel giving slightly better results than the others.

[3]Available at `http://reverb.cs.washington.edu/`

[4]See `reverb.cs.washington.edu/README_data.txt`

[5]Or, from a different perspective, changing *both* arguments with respect to the other fact, since facts used for this construction are picked at random.

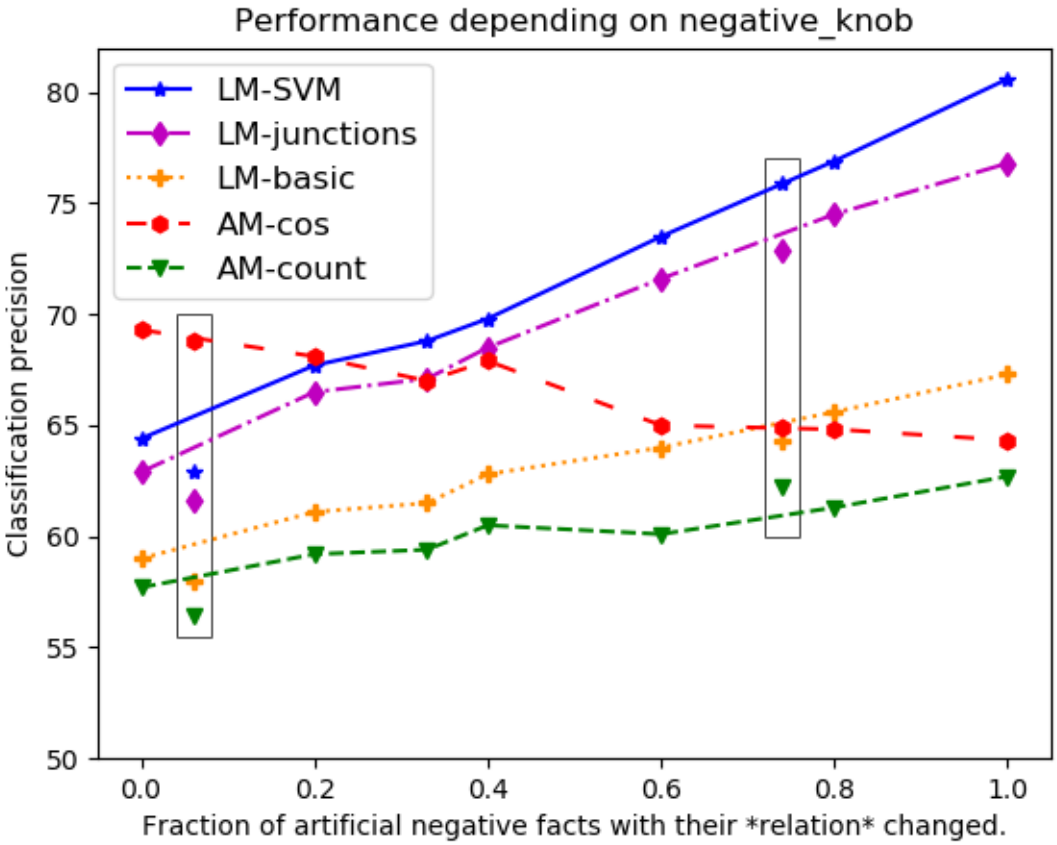

Figure 1: Performance depending on the proportion of negative examples where the *relation* was changed, rather than one of the arguments. 0.33 corresponds to picking at random, which we recommend. The method used by Angeli and Manning is not equivalent but corresponds to a value close to 0.75.

Angeli and Manning (2013) use a particular scheme : out of the 3 negative example candidates, they pick the one that has the largest cosine similarity with the original fact (a_1, r, a_2). This is with the stated purpose of training the classifier to discriminate between more similar examples, supposedly a better learning setup.

In practice, this means changing the relation in about 75% of cases, similarly to setting the knob at 0.75 in Figure 1 [6]. In practice, the fact most similar to the original tends to be the one with the swapped relation, because relational phrases are more often similar to each other than argument phrases. This in turn, we suppose, is because phrases are treated as bag of words and represented by their average word embedding, and relation phrases often share common verbs such as *be*, *make*, etc. and prepositions, whereas argument phrases are more distinct. Between $d(a_1, a'_1)$, $d(r, r')$ and $d(a_2, a'_2)$, $d(r, r')$ is the largest of the three in only 7% of cases (this choice of negative examples is the column of values at 0.07 on the x-axis).

Looking at Figure 1, we can see that **swapping the relation phrase makes the task easier** : except for AM-cos, all systems perform better, in a linear fashion. From a language modeling perspective, this is easy to explain : swapping the relation introduces two breaks inside the short sentence, where words may not fit together, instead of just one when an argument is replaced. When building artificial facts in this way, we recommend picking one of the 3 candidate negatives at random (i.e. setting the knob on 0.33). For more difficult learning tasks[7], further research could leverage other sources of information to produce better distractors as negative examples. This could be argument types, or using the fact that certain relations have only one correct value (e.g. *<person>, be born on, <date>*), so that differing values are known to be false.

4.4. Results on Manually Annotated Tuples

Our classification task up to now assumed all extracted facts to be correct, and all randomly-generated facts to be wrong. In practice this is not always the case, as some noise remains in the high-quality Reverb dataset, and some randomly assembled facts turn out to have interpretations that make them right, at least plausible. For instance (*a knight,*

[6] The difference is that our graph shows performance for a certain proportion of relation changes picked at random, whereas Angeli and Manning select a particular set of 75% relation changes. This is why points were made to be off the regression lines in the chart.

[7] For instance, distinguishing facts that are actually true in the world from facts that are false or sometimes false, or from facts that are merely plausible.

	Reverb-15M-manual
ArgSim	63.5
AM-count	47.7
AM-cos	57.5
LM-basic	52.6
LM-junctions	51.3

Table 2: Scores of unsupervised models on the manually annotated test set.

can turn into, a serious problem) would be true in medieval times, or in chess commentary.

Therefore, in an effort to come nearer the original task of recognizing impossible from plausible OIE extractions, we experiment with a small set of manually annotated facts. It is a delicate issue to decide what constitutes a correct extraction: see for instance Section 3.1 in (Stanovsky and Dagan, 2016). We annotated 430 tuples manually as:

- good extractions (45%) : capturing *some* information, and at least sometimes true. Some examples are (*Blackberry picking, is a great introduction to, foraging*), (*A new computer ; only costs around ; 500$*), and (*Blood pressure, is influenced by, dietary fibre*).

- unsure (30%) : typically only true in their original context which is lost, neither wrong nor useful in a vacuum. For instance (*Alfred, was against, garages*), (*Archaeology, answers this question with, confidence*), or (*Access ; is limited to ; official business*).

- incorrect extractions (25%) : nonsense or false, often due to upstream parser errors or noisy source text. E.g. (*5 Mts, Walk From, Wembley Stadium*), (*Atomic Kitten, released, the*) and (*A whole day ; set aside for ; literary pursuits*).

We annotate the tuples regardless of the sentences from which the facts were extracted : we label the facts as they stand and not the extraction process. The *unsure* labels (covering 25% of examples) were ignored and models had to classify correct and incorrect extractions. The negative examples were oversampled to balance the dataset.

The models used are the same as in Section 4.2., except that the SVM cannot be trained, as there is no training data (all facts are positive examples in the automatic task setup). Results are presented in Table 2. Among genuine extractions (all scored highly by Reverb's confidence function), current models have a hard time recognizing ill-formed or nonsensical extractions, and perform worse than on the automatically generated test set.

Some incorrect facts are scored highly because some particular pattern was often repeated over the web and systematically misinterpreted by the OIE system. One example is *Cross listed with <another class>.*, occuring frequently on certain university curriculum pages, from which Reverb extracts (*Cross, listed with,* e.g. *AIST2340*). This is correct from a language modeling perspective, even though it's a wrong fact.

4.5. Human performance

In an attempt to gauge the impact of false positives (genuine extractions by ReVerb that turn out to be erroneous) and false negatives (artificially assembled facts that turn out to be plausible), two human annotators, both NLP practitioners and including one author, manually performed the task on 200 facts. Half of them were not lemmatised. As in Table 1, the negative sampling method was that of (Angeli and Manning, 2013).

Both judges achieved just 80% accuracy at discriminating genuine and artificial facts, on both the lemmatised and unlemmatised versions of the task. Agreement was also 80%. Examples of highly ambiguous facts on which both annotators were mistaken include:

- (*Zaire ; will be maimed by ; betrayal*)
- (*Jean ; is a native of ; New York*)
- (*Peas ; here take advantage of ; ringtones*)
- (*Cooking school ; have changed a bit in ; the Los Angeles area*)

The first and third are genuine extractions (positive examples in the automated task) while the second and fourth were assembled at random (negative examples). Such facts constitute 10% of the test set.

Examples of facts on which annotators disagreed (one being mistaken) are:

- (*shimer college ; be establish in ; mt*)
- (*the north node ; will be in ; pisces*)
- (*Specific attention ; will be given to ; THE MAN*)
- (*Zoroastrianism ; is in even ; worse shape*)
- (*Kara ; is vice president of ; buying*)

Out of those five, only the third is artificial, and others are genuine extractions (the first two being lemmatised, as in the automatic evaluation). Such facts constitute 20% of the test set.

Overall, it is as if 60% of the automatically generated testset was reliably recognisable as genuine or artificial, humans performing no better than chance on the remaining 40% (hence the resulting 80% performance).

5. Conclusion

We revisit the task of judging the plausibility of a new candidate fact to extend a knowledge base, in the context of OIE — arbitrary relations between unrestricted noun phrases. Correctly assessing the validity of an unknown fact is highly valuable, both as a way to refine KBs built automatically, and to implicitly enhance finite stored knowledge for it to answer an order of magnitude more queries.

We propose a new baseline for this task, based on language modeling, which achieves state of the art performance. Indeed, archetypal correctly extracted information resembles short declarative sentences. We show that the way artificial negative examples are sampled has a large and robust impact on the difficulty of the task. We manually find genuine

yet incorrect extractions and show that while our system does capture some useful signal, picking up wrong extractions from a high quality dataset remains a challenging task. We examine the sort of facts that our model gets "right" despite the test set generation method being wrong, and the sort of facts on which it performs poorly. Future work extending the language model beyond off-the-shelf programs with named-entity recognition would improve its performance.

6. Acknowledgments

We would like to thank Fabrizio Gotti for his continuous help with annotation, alongside ever helpful comments. This work was supported by the Nuance Foundation.

7. Bibliographical References

Angeli, G. and Manning, C. D. (2013). Philosophers are mortal: Inferring the truth of unseen facts. In Julia Hockenmaier et al., editors, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 133–142. ACL.

Angeli, G., Premkumar, M. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *ACL*.

Bast, H. and Haussmann, E. (2013). Open information extraction via contextual sentence decomposition. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, pages 154–159. IEEE.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In C. J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

Del Corro, L. and Gemulla, R. (2013). Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 355–366, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Dong, X. L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610. Evgeniy Gabrilovich Wilko Horn Ni Lao Kevin Murphy Thomas Strohmann Shaohua Sun Wei Zhang Geremy Heitz.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Feng, J., Huang, M., Yang, Y., and Zhu, X. (2016). GAKE: graph aware knowledge embedding. In Nicoletta Calzolari, et al., editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 641–651. ACL.

García-Durán, A., Bordes, A., Usunier, N., and Grandvalet, Y. (2015). Combining two and three-way embeddings models for link prediction in knowledge bases. *CoRR*, abs/1506.00999.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, July. Association for Computational Linguistics.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38, November.

Li, X., Taheri, A., Tu, L., and Gimpel, K. (2016). Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, pages 1445–1455. Association for Computational Linguistics.

Mausam, Schmitz, M., Bart, R., Soderland, S., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., and Welling, J. (2015). Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *CoRR*, abs/1503.00759.

Pease, A., Niles, I., and Li, J. (2002). The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Riedel, S., Yao, L., Marlin, B. M., and McCallum, A. (2013). Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguis-*

tics (HLT-NAACL '13).

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., and Buetow, K. H. (2008). Pid: the pathway interaction database. *Nucleic acids research*, 37(suppl_1):D674–D679.

Stanovsky, G. and Dagan, I. (2016). Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas, November. Association for Computational Linguistics.

Stanovsky, G., Dagan, I., and Mausam. (2015). Open ie as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 303–308, Beijing, China, July. Association for Computational Linguistics.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA. ACM.

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In Lluís Màrquez, et al., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509. The Association for Computational Linguistics.

Toutanova, K., Lin, V., Yih, W., Poon, H., and Quirk, C. (2016). Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. *CoRR*, abs/1606.06357.

Wang, Q., Wang, B., and Guo, L. (2015). Knowledge base completion using embeddings and rules. In *IJCAI*, pages 1859–1866.

West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., and Lin, D. (2014). Knowledge base completion via search-based question answering. In *WWW*.

Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575.