IDEST: Learning a Distributed Representation for Event Patterns

Sebastian Krause[*]
LT Lab, DFKI
Alt-Moabit 91c
10559 Berlin, Germany
skrause@dfki.de

Enrique Alfonseca Katja Filippova Daniele Pighin
Google Research
Brandschenkestrasse 110
8810 Zurich, Switzerland
{ealfonseca,katjaf,biondo}@google.com

Abstract

This paper describes IDEST, a new method for learning paraphrases of event patterns. It is based on a new neural network architecture that only relies on the weak supervision signal that comes from the news published on the same day and mention the same real-world entities. It can generalize across extractions from different dates to produce a robust paraphrase model for event patterns that can also capture meaningful representations for rare patterns. We compare it with two state-of-the-art systems and show that it can attain comparable quality when trained on a small dataset. Its generalization capabilities also allow it to leverage much more data, leading to substantial quality improvements.

1 Introduction

Most Open Information Extraction (Open-IE) systems (Banko et al., 2007) extract textual relational patterns between entities automatically (Fader et al., 2011; Mausam et al., 2012) and optionally organize them into paraphrase clusters. These pattern clusters have been found to be useful for Question Answering (Lin & Pantel, 2001; Fader et al., 2013) and relation extraction (Moro & Navigli, 2012; Grycner & Weikum, 2014), among other tasks.

A related Open-IE problem is that of automatically extracting and paraphrasing **event patterns**: those that describe changes in the state or attribute values of one or several entities. An existing approach lo learn paraphrases of event patterns is to build on the following weak supervision signal: news articles that were published on the same day and mention the same entities should contain good paraphrase candidates. Two state-of-the-art event paraphrasing systems that are based on this assumption are NEWSSPIKE (Zhang & Weld, 2013) and HEADY (Alfonseca et al., 2013; Pighin et al., 2014).

These two systems have a lot in common, yet they have never been compared with each other. They have specific weak and strong points, and there are many ways in which they are substantially different:

- *Scope of generalization.* In NEWSSPIKE the paraphrase clusters are learned separately for each publication day and entity set, and the system cannot generalize across events of the same type involving different entities occurring on the same or on different days. For example, if the event verbs *has married* and *wed* appear in news about two entities A and B marrying, and *has married* and *tied the knot with* appear in news involving two different entities C and D, NEWSSPIKE is not able to infer that *wed* and *tied the knot with* are also paraphrases, unless a post-processing is done.

  HEADY overcomes this limitation thanks to a global model that learns event representations across different days and sets of entities. However, the global nature of the learning problem can incur into other drawbacks. First, training a global model is more costly and more difficult to parallelize. Second, relatively frequent patterns that erroneously co-occur with other patterns may have a negative impact on the final models, potentially resulting in noisier clusters. Lastly, low-frequency patterns are likely to be

[*] Work performed during an internship at Google

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1140–1149,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

discarded as noisy in the final model. Overall, HEADY is better at capturing paraphrases from the head of the pattern distribution, and is likely to ignore most of the long tail where useful paraphrases can still be found.

- *Simplifying assumptions.* We already mentioned that the two systems share a common underlying assumption, i.e., that good paraphrase candidates can be found by looking at news published on the same day and mentioning the same entities. On top of this, NEWSSPIKE also assumes that better paraphrases are reported around spiky entities, verb tenses may not differ, there is one event mention per discourse, and others. These restrictions are not enforced by HEADY, where the common assumption is indeed even relaxed across days and entity sets.

- *Annotated data.* NEWSSPIKE requires hand-annotated data to train the parameters of a supervised model that combines the different heuristics, whereas HEADY does not need annotated data.

This paper describes IDEST, a new method for learning paraphrases of event patterns that is designed to combine the advantages of these two systems and compensate for their weaknesses. It is based on a new neural-network architecture that, like HEADY, only relies on the weak supervision signal that comes from the news published on the same day, requiring no additional heuristics or training data. Unlike NEWSSPIKE, it can generalize across different sets of extracted patterns, and each event pattern is mapped into a low-dimensional embedding space. This allows us to define a neighborhood around a pattern to find the ones that are closer in meaning.

IDEST produces a robust global model that can also capture meaningful representations for rare patterns, thus overcoming one of HEADY's main limitations. Our evaluation of the potential trade-off between local and global paraphrase models shows that comparably good results to NEWSSPIKE can be attained without relying on supervised training. At the same time, the ability of IDEST to produce a global model allows it to benefit from a much larger news corpus.

2 Related work

Relational Open-IE In an early attempt to move away from domain-specific, supervised IE systems, Riloff (1996) attempted to automatically find relational patterns on the web and other unstructured resources in an open domain setting. This idea has been further explored in more recent years by Brin (1999), Agichtein & Gravano (2000), Ravichandran & Hovy (2002) and Sekine (2006), among the others. Banko et al. (2007) introduced Open-IE and the TEXTRUNNER system, which extracted binary patterns using a few selection rules applied on the dependency tree. More recent systems such as REVERB (Fader et al., 2011) and OLLIE (Mausam et al., 2012) also define linguistically-motivated heuristics to find text fragments or dependency structures that can be used as relational patterns.

A natural extension to the previous work is to automatically identify which of the extracted patterns have the same meaning, by producing either a hard or a soft clustering. Lin & Pantel (2001) use the mutual information between the patterns and their observed slot fillers. Resolver (Yates & Etzioni, 2007) introduces a probabilistic model called the Extracted Shared Property (ESP) where the probability that two instances or patterns are paraphrases is based on how many properties or instances they share. USP (Poon & Domingos, 2009) produces a clustering by greedily merging the extracted relations. Yao et al. (2012) employ topic models to learn a probabilistic model that can capture also the ambiguity of polysemous patterns. More recent work also organizes patterns in clusters or taxonomies using distributional methods on the pattern contexts or entities extracted (Moro & Navigli, 2012; Nakashole et al., 2012), or implicitly clusters relational text patterns via the learning of latent feature vectors for entity tuples and relations, in a setting similar to knowledge-base completion (Riedel et al., 2013).

A shared difficulty for systems that cluster patterns based on the arguments they select is that it is very hard for them to distinguish between identity and entailment. If one pattern entails another, both are likely to be observed in the corpus involving the same entity sets. A typical example illustrating this problem is the two patterns e_1 *married* e_2 and e_1

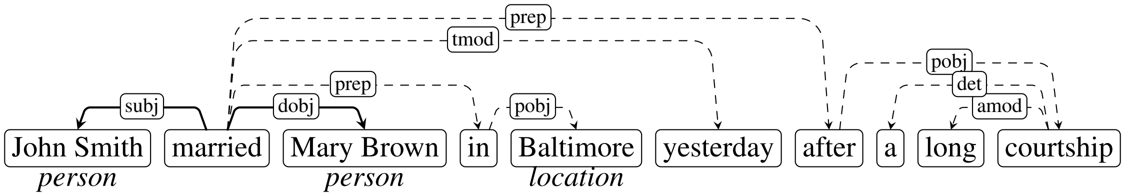

Figure 1: Example sentence, and extraction (the nodes connected through solid dependency edges).

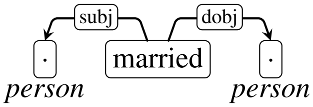

Figure 2: Example pattern that encodes a wedding event between two people.

dated e_2, which can be observed involving the same pairs of entities, but which carry a different meaning. As discussed below, relying on the temporal dimension (given by the publication date of the input documents) is one way to overcome this problem.

Event patterns and Open-IE Although some earlier work uses the temporal dimension of text as filters to improve precision of relational pattern clusters, NEWSSPIKE (Zhang & Weld, 2013) and HEADY (Alfonseca et al., 2013; Pighin et al., 2014) fully rely on it as its main supervision signal. In order to compare the two approaches, we will start by defining some terms:

- An **event pattern** encodes an expression that describes an event. It can be either a linear surface pattern or a lexico-syntactic pattern, and can possibly include entity-type restrictions on the arguments. For example, Figure 2 represents a binary pattern that corresponds to a wedding event between two people.

- An **extraction** is a pattern instance obtained from an input sentence, involving specific entities. For example, the subgraph represented with solid dependency edges in Figure 1 is an extraction corresponding to the pattern in Figure 2.

- An **Extracted Event Candidate Set** (EEC-Set (Zhang & Weld, 2013), or just EEC for brevity) is the set of extractions obtained from news articles published on the same day, and involving the same set of entities.

- Two extractions are **co-occurrent** if there is at least one EEC that contains both of them.

NEWSSPIKE produces extractions from the input documents using REVERB (Fader et al., 2011). The EECs are generated from the titles and all the sentences of the first paragraph of the documents published on the same day. From each EEC, potentially one paraphrase cluster may be generated. The model is a factor graph that captures several additional heuristics. Integer Lineal Programming (ILP) is then used to find the Maximum a Posteriori (MAP) solution for each set of patterns, and model parameters are trained using a labeled corpus that contains 500 of these sets.

Regarding HEADY, it only considers titles and first sentences for pattern extraction and trains a two-layer Noisy-OR Bayesian Network, in which the hidden nodes represent possible event types, and the observed nodes represent the textual patterns. A maximum-likelihood model is the one in which highly co-occurring patterns are generated by the same latent events. The output is a global soft clustering, in which two patterns may also be clustered together even if they never co-occur in any EEC, as long as there is a chain of co-occurring patterns generated by the same hidden node. HEADY was evaluated using three different extraction methods: a heuristic-based pattern extractor, a sentence compression algorithm and a memory-based method.

While this model produces a soft clustering of patterns, HEADY was evaluated only on a *headline generation* task and not intrinsically w.r.t. the quality of the clustering itself.

Neural networks and distributed representations Another related field aims to learn continuous vector representations for various abstraction levels of natural language. In particular the creation of so-called word embeddings has attracted a lot of attention in the past years, often by implementing neural-network language models. Prominent examples include the works by Bengio et al. (2003) and Mikolov et al. (2013), with the skip-gram model of the latter providing a basis for the vector representations learned in our approach.

Also closely related to IDEST are approaches which employ neural networks capable of handling word sequences of variable length. For example, Le & Mikolov (2014) extend the architectures of Mikolov et al. (2013) with artificial paragraph tokens, which accumulate the meaning of words appearing in the respective paragraphs.

In contrast to these shallow methods, other approaches employ deep multi-layer networks for the processing of sentences. Examples include Kalchbrenner et al. (2014), who employ convolutional neural networks for analyzing the sentiment of sentences, and Socher et al. (2013), who present a special kind of recursive neural network utilizing tensors to model the semantics of a sentence in a compositional way, guided by the parse tree.

A frequent issue with the deeper methods described above is the high computational complexity coming with the large numbers of parameters in a multi-layer neural network or in the value propagation in unfolded recursive neural networks. To circumvent this problem, our model is inspired by Mikolov's simpler skip-gram model, as described below.

3 Proposed model

Similarly to HEADY and NEWSSPIKE, our model is also based on the underlying assumption that if sentences from two news articles were published on the same day and mention the same entity set, then they are good paraphrase candidates. The main novelty is in the way we train the paraphrase model from the source data. We propose a new neural-network architecture which is able to learn meaningful distributed representations of full patterns.

3.1 Skip-gram neural network

The original Skip-gram architecture (Mikolov et al., 2013) is a feed-forward neural network that is trained on distributional input examples, by assuming that each word should be able to predict to some extent the other words in its context. A skip-gram architecture consists of:

1. An input layer, usually represented as a *one-of-V* or *one-hot-spot* layer. This layer type has as many input nodes as the vocabulary size. Each training example will activate exactly one input node corresponding to the current word w_i, and all the other input nodes will be set to zero.

2. A first hidden layer, the embedding or projection layer, that will learn a distributed representation for each possible input word.

3. Zero or more additional hidden layers.

4. An output layer, expected to predict the words in the context of w_i: $w_{i-K}, ..., w_{i-1}, w_{i+1}, ..., w_{i+K}$.

In practice, when training based on this architecture, the network converges towards representing words that appear in similar contexts with vectors that are close to each other, as close vectors will produce a similar distribution of output labels in the network.

3.2 IDEST neural network

Figure 3 shows the network architecture we use for training our paraphrase model in IDEST. In our case, the input vocabulary is the set of N unique event patterns extracted from text, and our supervision signal is the co-occurrence of event patterns in EECs. We set the input to be a *one-hot-spot* layer with a dimensionality of N, and for each pair of patterns that belong to the same EECs, we will have these patterns predict each other respectively, in two separate training examples. The output layer is also a *one-of-V* layer, because for each training example only one output node will be set to 1, corresponding to a co-occurring pattern.

After training, if two patterns P_i and P_j have a large overlap in the set of entities they co-occur with, then they should be mapped onto similar internal representations. Note that the actual entities are only used for EEC construction, but they do not play a role in the training itself, thus allowing the network to generalize over specific entity instantiations. To exemplify, given the two EECs {"*[Alex] married [Leslie]*", "*[Leslie] tied the knot with*

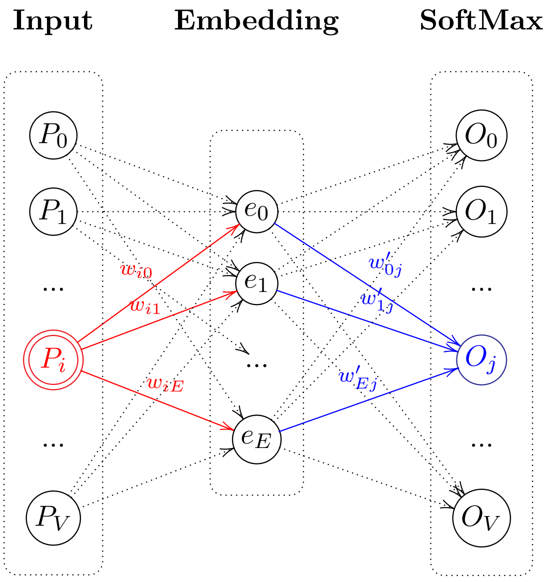

Figure 3: Model used for training. V is the total number of unique patterns, which are used both in the *one-of-V* input and output. E is the dimensionality of the embedding space.

[Alex]"} and {"*[Carl] and [Jane] wed*", "*[Carl] married [Jane]*"}, IDEST could learn an embedding space in which "*[Per] tied the know with [Per]*" and "*[Per] and [Per] wed*" are relatively close, even though the two patterns never co-occur in the same EEC. This is possible because both patterns have been trained to predict the same pattern {"*[Per] married [Per]*"}.

Reported representations of word embeddings typically use between 50 and 600 dimensions (Mikolov et al., 2013; Levy & Goldberg, 2014). For our pattern embeddings we have opted for an embedding layer size of 200 nodes. We also experimented with larger sizes and with adding more intermediate hidden layers, but while the added cost in terms of training time was substantial we did not observe a significant difference in the results.

4 Experimental settings

4.1 Pattern extraction methods used

In previous work we can find three different pattern extraction methods from a sentence:

- *Heuristic-based*, where a number of handwritten rules or regular expressions based on part-of-speech tags or dependency trees are used to select the most likely pattern from the source sentence (Fader et al., 2011; Mausam et al., 2012; Alfonseca et al., 2013).

- *Sentence compression*, which takes as input the original sentence and the entities of interest and produces a shorter version of the sentence that still includes the entities (Pighin et al., 2014).

- *Memory-based*, that tries to find the shortest reduction of the sentence that still includes the entities, with the constraint that its lexico-syntactic structure has been seen previously as a full sentence in a high-quality corpus (Pighin et al., 2014).

It is important to note that the final purpose of the system may impact the decision of which extraction method to choose. Pighin et al. (2014) use the event models to generate headlines, and using the memory-based method resulted in more grammatical headlines at the cost of coverage. If the purpose of the patterns is information extraction for knowledge base population, then the importance of having well-formed complete sentences as patterns becomes less obvious, and higher coverage methods become more attractive. For these reasons, in this paper we focus on the first two approaches, which are very well-established and can produce high-coverage output. More specifically, we use REVERB extractions and a statistical compression model trained on (sentence, compression) pairs implemented after Filippova & Altun (2013).

4.2 Generating clusters from the embedding vectors

IDEST does not produce a clustering like NEWSSPIKE and HEADY, so in order to be able to compare against them we have used the algorithm described in Figure 4 to build paraphrase clusters from the pattern embeddings. Given a similarity threshold on the cosine similarity of embedding vectors, we start by sorting the patterns by extraction frequency and proceed in order along the sorted vector by keeping the most similar pattern of each. Used patterns are removed from the original set to make sure that a pattern is not added to two clusters at the same time.

```
function ComputeClusters($P$, $\theta$)
    $Result = \{\}$
    SortByFrequency($P$)
    while $|P| > 0$ do
        $p =$ Pop($P$)                    ▷ Take highest-frequency pattern
        $C_p = \{p\}$                     ▷ Initialize cluster around $p$
        $N =$ Neighbors($p, P, \theta$)   ▷ $n \in P, sim(n,p) > \theta$
        for all $n \in N$ do
            $C_p = C_p \cup \{n\}$
            Remove($P, n$)               ▷ Remember $n$ has been used
        $Result = Result \cup \{C_p\}$
    return $Result$
```

Figure 4: Pseudocode of the algorithm for producing a clustering from the distributed representation of the extracted patterns. P is the set of extracted patterns, and θ is the similarity threshold to include two patterns in the same cluster.

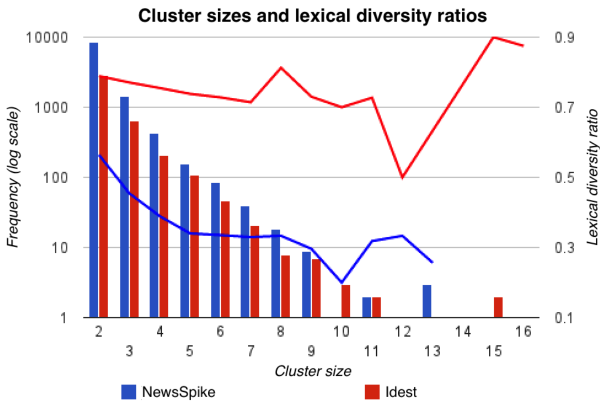

Figure 5: Cluster size (log-scale) and ratio of unique verb lemmas in the clusters generated from NewsSpike and Idest with the ReVerb extractions as input.

5 Evaluation results

This section opens with a quantitative look at the clusterings obtained with the different methods to understand their implications with respect to the distribution of event clusters and their internal diversity. In 5.2, we will complement these figures with the results of a manual quality evaluation.

5.1 Quantitative analysis

5.1.1 NewsSpike vs. Idest-ReV-NS

This section compares the clustering models that were output by NewsSpike and Idest when using the same set of extractions, to evaluate the performance of the factor graph-based method and the neural-network method on exactly the same EECs. We have used as input the dataset released by Zhang & Weld (2013)[1], which contains 546,713 news articles, from which 2.6 million ReVerb extractions were reportedly produced. 84,023 of these are grouped into the 23,078 distributed EECs, based on mentions of the same entities on the same day. We compare here the released output clusters from NewsSpike and a clustering obtained from a Idest-based distributed representation trained on the same EECs.

Figure 5 shows a comparative analysis of the two sets of clusters. As can be seen, Idest generates somewhat fewer clusters for every cluster size than NewsSpike. We have also computed a *lexical diversity ratio*, defined as the percentage of root-verb lemmas in a cluster that are unique. This metric captures whether a cluster mainly contains the same verb with different inflections or modifiers, or whether it contains different predicates. The figure shows that Idest generates clusters with much more lexical diversity. These results make sense intuitively, as a global model should be able to produce more aggregated clusters by merging patterns originating from different EECs, resulting in fewer clusters with a higher lexical diversity. A higher lexical diversity may be a signal of richer paraphrases or noisier clusters. The manual evaluation in Section 5.2 will address this issue by comparing the quality of the clusterings.

5.1.2 NewsSpike vs. Idest-Comp-NS

Figure 6 compares NewsSpike's clusters against Idest clusters obtained using sentence compression instead of ReVerb for extracting patterns. Both systems were trained on the same set of input news. Using sentence compression, the total number of extracted patterns was 321,130, organized in 41,740 EECs. We can observe that Idest produced larger clusters than NewsSpike. For cluster sizes larger or equal to 4, this configuration of Idest produced more clusters than NewsSpike. At the same time, lexical diversity remained consistently on much higher levels, well over 60%.

5.1.3 Idest-Comp-NS vs. Idest-Comp-All

In order to evaluate the impact of the size of training data, we produced a clustering from embedding vectors trained from a much larger dataset. We used

[1] http://www.cs.washington.edu/node/9473

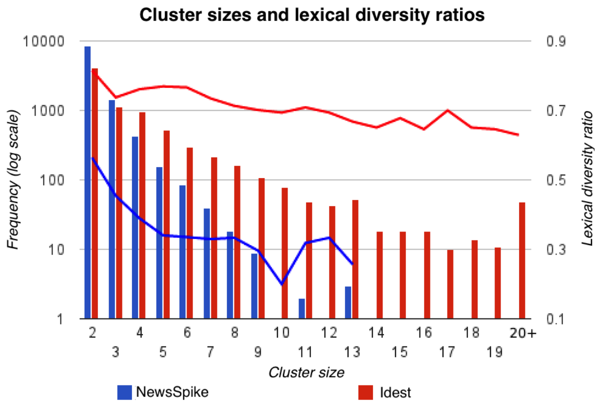

Figure 6: Cluster size (log-scale) and ratio of unique verb lemmas in the clusters generated from NEWSSPIKE and IDEST with compression-based pattern extraction.

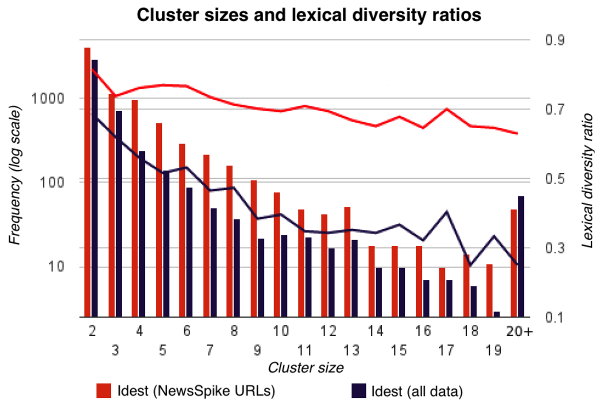

Figure 7: Cluster size (log-scale) and ratio of unique verb lemmas in the clusters generated from IDEST with compression-based pattern extraction, using only the 500,000 NEWSSPIKE articles, or the large dataset.

our own crawl of news collected between 2008 and 2014. Using sentence compression, hundreds of millions of extractions have been produced.

In order to keep the dataset at a reasonable size, and aiming at producing a model of comparable size to the other approaches, we applied a filtering step in which we removed all the event patterns that were not extracted at least five times from the dataset. After this filtering, 28,014,423 extractions remained, grouped in 8,340,162 non-singleton EECs.

Figure 7 compares the resulting clusterings. In the all-data setting, clusters were generally smaller and showed less lexical variability. We believe that this is due to the removal of the long tail of low-frequency and noisy patterns. Indeed, while high lexical variability is desirable it can also be a sign of noisy, unrelated patterns in the clusters. The cohesiveness of the clusters, which we will evaluate in Section 5.2, must also be considered to tell constructive and destructive lexical variability apart.

5.1.4 HEADY

HEADY produces a soft-clustering from a generative model, and expects the maximum number of clusters to be provided beforehand. The model then tries to approximate this number. In our experiments, 5,496 clusters were finally generated. One weak point of HEADY, mentioned above, is that low-frequency patterns do not have enough evidence and Noisy-OR Bayesian Networks tend to discard them; in our experiments, only 4.3% of the unique extracted patterns actually ended up in the final model.

5.2 Qualitative analysis

The clusters obtained with different systems and dataset have been evaluated by five expert raters with respect to three metrics, according to the following rating workflow:

1. The rater is shown the cluster, and is asked to annotate which patterns are meaningless or unreadable[2]. This provides us with a **Readability** score, which measures at the same time the quality of the extraction algorithm and the ability of the method to filter out noise.

2. The rater is asked whether there is a majority theme in the cluster, defined as having at least half of the readable patterns refer to the same real-world event happening. If the answer is *No*, the cluster is annotated as noise. We call this metric **Cohesiveness**.

3. If a cluster is cohesive, the rater is finally asked to indicate which patterns are expressing the main theme, and which ones are unrelated to it. The third metric, **Relatedness**, is defined as the percentage of patterns that are related to the main cluster theme. All the patterns in a non-cohesive cluster are automatically marked as unrelated.

[2] In the data released by NewsSpike, REVERB patterns are lemmatized, but the original inflected sentences are also provided. We have restored the original inflection of all the words to make those patterns more readable for the raters.

System	Ext	Data	Size	Coh(%)	Rel(%)	Read(%)
HEADY	Comp	All	12.66^{bcd}	34.40$^{!}$	27.70$^{!}$	60.70
NEWSSPIKE	ReV	NS	3.40$^{!}$	56.20^{ac}	66.42^{acd}	56.66
IDEST	ReV	NS	3.62^{b}	40.00	47.10^{a}	65.16^{b}
IDEST	Comp	NS	5.54^{bc}	50.31^{ac}	46.58^{a}	66.04^{b}
IDEST	Comp	All	44.09*	87.93*	68.28^{acd}	80.13*

Table 1: Results of the manual evaluation, averaged over all the clusters produced by each configuration listed. **Extraction algorithms**: *ReV* = REVERB; *Comp* = Compression; **Data sets**: *NS* = NewsSpike URLs; *All* = news 2008-2014. **Quality metrics**: *Size*: average cluster size; *Coh*: cohesiveness; *Rel*: relatedness; *Read*: readability. **Statistical significance**: ^{a}: better than HEADY; ^{b}: better than NEWSSPIKE; ^{c}: better than IDEST-ReV-NS; ^{d}: better than IDEST-Comp-NS; *: better than all others; $^{!}$: worse than all others (0.95 confidence intervals, bootstrap resampling).

The inter-annotator agreement on the three metrics, measured as the intraclass correlation (ICC), was strong (Cicchetti, 1994; Hallgren, 2012). More precisely, the observed ICC scores (with 0.95 confidence intervals) were 0.71 [0.70, 0.72] for cohesiveness, 0.71 [0.70, 0.73] for relatedness and 0.66 [0.64, 0.67] for readability.

For the evaluation, from each model we selected enough clusters to achieve an overall size (number of distinct event patterns) comparable to NEWSSPIKE's. For HEADY and IDEST, the stopping condition in Figure 4 was modified accordingly.

Table 1 shows the outcome of the annotation. As expected, using a global model (that can merge patterns from different EECs into single clusters) and using the whole news dataset both led to larger clusters. At the same time, we observe that using REVERB extractions generally led to smaller clusters. This is probably because REVERB produced fewer extractions than sentence compression from the same input documents.

On REVERB extractions, NEWSSPIKE outperformed IDEST in terms of cohesiveness and relatedness, but NEWSSPIKE's lowest cluster size and lexical diversity makes it difficult to prefer any of the two models only w.r.t. the quality of the clusters. On the other hand, the patterns retained by IDEST-ReV-NS were generally more readable (65.16 vs. 56.66).

On the same original news data, using IDEST with sentence compression produced comparable results to IDEST-ReV-NS, Cohesiveness being the only metric that improved significantly.

More generally, in terms of readability all the models that rely on global optimization (i.e., all but NEWSSPIKE) showed better readability than NEWSSPIKE, supporting the intuition that global models are more effective in filtering out noisy extractions. Also, the more data was available to IDEST, the better the quality across all metrics. IDEST model using all data, i.e, IDEST-Comp-All, was significantly better (with 0.95 confidence) than all other configurations in terms of cluster size, cohesiveness and pattern readability. Pattern relatedness was higher, though not significantly better, than NEWSSPIKE, whose clusters were on average more than ten times smaller.

We did not evaluate NEWSSPIKE on the whole news dataset. Being a local model, extending the dataset to cover six years of news would only lead to many more EECs, but it would not affect the reported metrics as each final cluster would still be generated from one single EEC.

It is interesting to see that, even though they were trained on the same data, IDEST outperformed HEADY significantly across all metrics, sometimes by a very large margin. Given the improvements on cluster quality, it would be interesting to evaluate IDEST performance on the headline-generation task for which HEADY was initially designed, but we leave this as future work.

6 Conclusions

We described IDEST, a new approach based on neural networks to map event patterns into an embedding space. We show that it can be used to construct high quality pattern clusters based on neighborhood in the embedding space. On a small dataset, IDEST produces comparable results to NEWSSPIKE, but its main strength is in its ability to generalize extractions into a single global model. It scales to hundreds of millions of news, leading to larger clusters of event patterns with significantly better coherence and readability. When compared to HEADY, IDEST outperforms it significantly on all the metrics tried.

Acknowledgments

The first author was partially supported by the German Federal Ministry of Education and Research, project ALL SIDES (contract 01IW14002).

References

Agichtein, E. & L. Gravano (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pp. 85–94.

Alfonseca, E., D. Pighin & G. Garrido (2013). HEADY: News headline abstraction through event pattern clustering. In *Proc. of ACL-13*, pp. 1243–1253.

Banko, M., M. J. Cafarella, S. Soderland, M. Broadhead & O. Etzioni (2007). Open information extraction from the Web. In *Proc. of IJCAI-07*, pp. 2670–2676.

Bengio, Y., R. Ducharme & P. Vincent (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Brin, S. (1999). Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pp. 172–183. Springer.

Cicchetti, D. V. (1994). Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological Assessment*, 6(4):284.

Fader, A., S. Soderland & O. Etzioni (2011). Identifying relations for open information extraction. In *Proc. of EMNLP-11*, pp. 1535–1545.

Fader, A., L. S. Zettlemoyer & O. Etzioni (2013). Paraphrase-driven learning for open question answering. In *Proc. of ACL-13*, pp. 1608–1618.

Filippova, K. & Y. Altun (2013). Overcoming the lack of parallel data in sentence compression. In *Proc. of EMNLP-13*, pp. 1481–1491.

Grycner, A. & G. Weikum (2014). Harpy: Hypernyms and alignment of relational paraphrases. In *Proc. of COLING-14*, pp. 2195–2204.

Hallgren, K. A. (2012). Computing inter-rater reliability for observational data: An overview and tutorial. *Tutorials in quantitative methods for psychology*, 8(1):23.

Kalchbrenner, N., E. Grefenstette & P. Blunsom (2014). A convolutional neural network for modelling sentences. In *Proc. of ACL-14*.

Le, Q. & T. Mikolov (2014). Distributed representations of sentences and documents. In *Proc. of ICML-14*.

Levy, O. & Y. Goldberg (2014). Linguistic regularities in sparse and explicit word representations. In *Proc. of CoNLL-14*.

Lin, D. & P. Pantel (2001). Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Mausam, M. Schmitz, R. Bart, S. Soderland & O. Etzioni (2012). Open language learning for information extraction. In *Proc. of EMNLP-12*, pp. 523–534.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado & J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119.

Moro, A. & R. Navigli (2012). Wisenet: Building a wikipedia-based semantic network with ontologized relations. In *Proc. of CIKM-12*, pp. 1672–1676.

Nakashole, N., G. Weikum & F. Suchanek (2012). Patty: a taxonomy of relational patterns with semantic types. In *Proc. of EMNLP-12*, pp. 1135–1145.

Pighin, D., M. Colnolti, E. Alfonseca & K. Filippova (2014). Modelling events through memory-based, Open-IE patterns for abstractive summarization. In *Proc. of ACL-14*, pp. 892–901.

Poon, H. & P. Domingos (2009). Unsupervised semantic parsing. In *Proc. of EMNLP-09*, pp. 1–10.

Ravichandran, D. & E. H. Hovy (2002). Learning surface text patterns for a question answering system. In *Proc. of ACL-02*, pp. 41–47.

Riedel, S., L. Yao, B. M. Marlin & A. McCallum (2013). Relation extraction with matrix factorization and universal schemas. In *Proc. of HLT-NAACL-13*.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proc. of AAAI-96*, pp. 1044–1049.

Sekine, S. (2006). On-demand information extraction. In *Proc. of COLING-ACL-06 Poster Session*, pp. 731–738.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng & C. Potts (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP-13*.

Yao, L., S. Riedel & A. McCallum (2012). Unsupervised relation discovery with sense disambiguation. In *Proc. of ACL-12*, pp. 712–720.

Yates, A. & O. Etzioni (2007). Unsupervised resolution of objects and relations on the Web. In *Proc. of NAACL-HLT-07*, pp. 121–130.

Zhang, C. & D. S. Weld (2013). Harvesting parallel news streams to generate paraphrases of event relations. In *Proc. of EMNLP-13*, pp. 1776–1786.