NIST TAC SM-KBP 2019 System Description: JHU/UR Framework

Yunmo Chen, Seth Ebner, Tongfei Chen, Patrick Xia, Elias Stengel-Eskin
Tzu-Ray Su, J. Edward Hu, Nils Holzenberger, Ryan Culkin, Craig Harman
Max Thomas, Thomas Lippincott, Aaron Steven White†, Kyle Rawlins, Benjamin Van Durme
Johns Hopkins University; †University of Rochester

Abstract

We designed and constructed a pipeline system for the Streaming Multimedia Knowledge Base Population (SM-KBP) 2019 evaluation. Our pipeline consists of a series of Information Extraction and Machine Translation components, supporting the population of knowledge graphs from a multilingual corpus (see Figure 1). Some of our key contributions include: a new dataset supporting event argument linking across multiple sentences; a novel algorithm for predicting such linkages, even when arguments are not co-referent with mentions in the same sentence as the event trigger; and a novel algorithm for supporting hierarchical typing of events, relations and entities.

1 Introduction

The Streaming Multimedia Knowledge Base Population (SM-KBP) task aims at constructing knowledge graph that supports structured queries against unstructured multi-media sources. In the 2019 evaluation there were three main subtasks: information extraction and population of knowledge elements (KE) (TA1), knowledge graph construction (TA2), hypotheses generation from KB (TA3). Here we are focussed on the TA1 task.

In the evaluation, participants were required to process a set of independent documents under an existing ontology, first without and then with some notion of downstream information needs (referred to as a *hypothesis*). For a each document we sequentially performed a series of tasks: Entity Mention Detection (EMD), Coreference Resolution, Event Trigger Detection, Entity / Event / Relation Typing, and Argument Linking. In addition, as documents might be in Russian or Ukrainian, we employed techniques such as Language Identification, Machine Translation, and Word Alignment, which enables us to pivot on the models trained on English corpora and project the discovered text span offsets back to the original documents. For hypothesis-based population in TA1b, we followed a strategy similar to our AIDA M9 approach, incorporating a textual similarity matching model for Entity Linking. The final processed results were converted to AIDA Interchange Format (AIF). Our system is depicted in Figure 1.

We experimented with different encoders, including BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018), and decided to use BERT as the main source of extraction features.

To combat data sparsity in the released practice data from LDC ahead of the evaluation, we developed a new annotated resource based on the AIDA Annotation Ontology (as of Spring 2019).

2 Multi-Sentence Events and Relations Dataset

We constructed a crowd-sourced dataset with 9,100 AIDA event and relation annotations. Each data point consists of a typed *trigger* span and 0 or more typed *argument* spans in an English document. A trigger span is a word or phrase that evokes a certain event or relation type in the context of a document (e.g. "pledge" may evoke the "Contact.CommitmentPromiseExpressIntent.Broadcast" event in certain contexts), while argument spans denote participant types in the event or relation (e.g. the "Communicator" or the "Recipient"). Both event and arguments spans are token-level (start, end) offsets into a tokenized text document.

Typically, event and relation datasets force argument spans to be in the same sentence as the trigger span, but we present annotators with a *multi-sentence* window: the sentence containing the trigger span, some number of sentences before, and some number sentences after. The annotator can then select argument spans anywhere inside of the context window. In practice, we showed annotators five sentences: the trigger sentence, two sentences before, and two after.

We used Reddit, a popular internet forum, to identify a suitable collection of texts that were likely to contain AIDA events and relation mentions. On Reddit, users post submissions containing links to news articles, images, videos, or other kinds of documents, and other users may then vote or comment on posted submissions. We considered news articles matching the following criteria:

1. Article was posted to the r/politics sub-forum

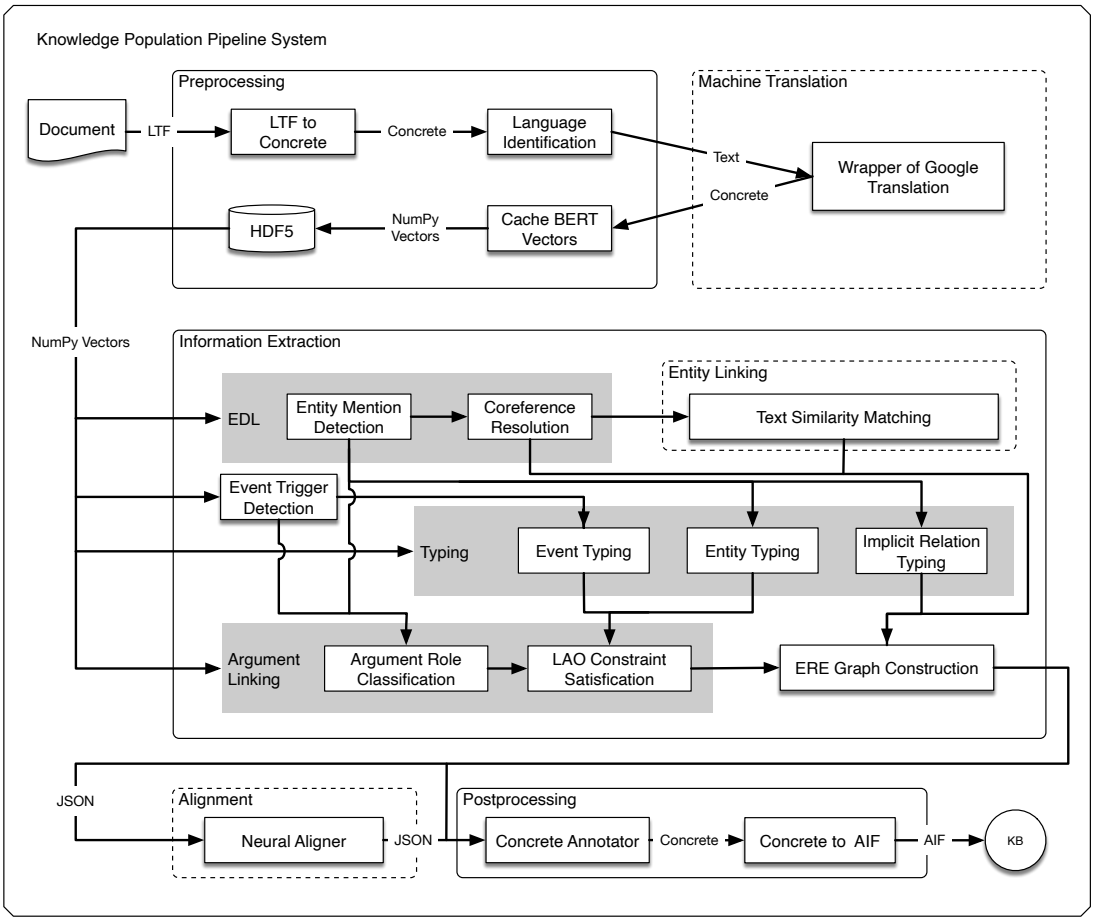

Figure 1: Data flow for populating knowledge graphs from a document. The processes within the dashed rectangles are optional and will be applied to certain inputs; Machine Translation and Alignment modules are activated only when the language is not English; Entity Linking module is only activated for TA1b evaluation.

2. Article contained at least one mention of the string "Russia"
3. Article had at least 25 comments
4. Article was posted between January and October 2016

By considering politically-themed articles containing the word "Russia", the texts tended to describe geopolitical events and relations like the ones in the AIDA ontology. In order to filter out low-quality, fake, or disreputable news articles, we only considered submissions that had generated at least 25 comments. After applying these criteria we identified approximately 12,000 news articles, each with an average of approximately 40 sentences.

Next, we manually constructed a mapping from each AIDA event and relation (sub-(sub-))type to a small list of lexical units (LUs) likely to evoke that type. This mapping was high-precision, low-recall, in the sense that for a given (Type, WordList) pair, the LUs in the WordList were all likely to evoke the Type, but the WordList could lack many LUs that also evoked the Type. Each AIDA type had 3.89 manually-curated LUs on average.

Using the AIDA-to-LU mapping we performed a soft match between every LU in the mapping and every word in our collection of texts in order to select candidate sentences with respect to each event and relation type. In the soft matching procedure, we lemmatized all words to their base form, removing any inflectional endings, and lower-cased the strings in order to get a high-recall set of candidate sentences. This matching procedure returned approximately 94,000 candidates, which we then balanced at the lexical unit level, i.e. we sampled the same number of candidate sentences for each lexical unit.

Candidate sentences were then manually vetted to ensure that they matched their associated proposed event or relation type. Each vetting task given to annotators contained an event or relation definition and several candidate sentences, each with a highlighted LU. Annotators were asked to judge how well each highlighted LU, in the context of its sentence, matched the provided event or relation definition. They were also asked to assess the factuality of the sentence - whether the event or relation actually happened. We collected judgments on approximately 17,500 candidate sentences.

Of the 17,500 candidate sentences, 52% were determined to match their provided definition and have positive factuality, giving us 9,100 sentences, each with a highlighted LU known to evoke a given event or relation type. Using these sentences we then collected multi-sentence argument annotations, presenting annotators with a 5-sentence window with two sentences of context before the sentence with the trigger, and two sentences after. On average, we have 66 full annotations (trigger and argument) per AIDA type.

3 Models and Experiments

3.1 Machine Translation

For our information extraction systems, it was necessary to translate Ukrainian (Uk) and Russian (Ru) into English (En). For the purposes of our alignment system (see Section 3.7), we also needed models that could translate Ukrainian and Russian to English. We built models for all 4 settings, and compared to Google translate's performance.

3.1.1 Ukrainian - English

Uk-En parallel corpora are quite scarce, in particular for the news domain, which was most relevant to the evaluation scenario. We relied on two datasets. The first one, OpenSubtitles [1], is as far as we know the largest openly available Uk-En parallel corpus, with 878k sentences. The sentences were extracted from movie subtitles (Lison and Tiedemann, 2016). We used this corpus as an initial training set for our Uk→En and En→Uk models. The second corpus we used is the SUMMA corpus (Germann et al., 2019). The data was scraped from the beginning of 2016 to November 2018, from two Ukrainian news sites, censor.net (75%) and UNIAN (25%), making the SUMMA corpus very relevant domain-wise. Out of the 88k sentences that the unprocessed SUMMA corpus contains, we randomly set aside 5% as a development set, which we report performance on. The remaining 95% were used to fine-tune models trained on OpenSubtitles.

We used the Sockeye toolkit (Hieber et al., 2017) to build RNN-based and Transformer-based sequence-to-sequence models, together with Byte-Pair Encoding (BPE) (Sennrich et al., 2015). The best models for both Uk→En and En→Uk used 2 LSTM layers with 512 hidden units, 30k BPE operations, 128-dimensional embeddings for subword units, an initial learning rate of 0.001, a batch size of 64 and dot product attention.

The performance of our models on the SUMMA dev set are reported in the top of Table 1 under "NMT", together with the performance of Google translate. Given that we outperformed Google translate, we used our own models in the evaluation.

[1] http://opus.nlpl.eu/download.php?f=OpenSubtitles2018/en-uk.txt.zip

Table 1: Machine translation performance (BLEU)

	NMT	Google translate
Uk → En	28.2	24.4
En → Uk	22.5	19.5
Ru → En	28.4	-
En → Ru	27.0	32.4

3.1.2 Russian - English

We used the 25M sentences from the WMT'17 news translation challenge [2] to build En→Ru and Ru→En models. The corpus in question contains the Common Crawl corpus, News Commentary v12, Yandex Corpus, Wiki Headlines, and UN Parallel Corpus V1.0. The WMT'17 news corpus is thus appropriate domain-wise. We report performance on the WMT'17 test set.

Models were built using the trained with the Sockeye toolkit and BPE. Our model used a Transformer-based sequence-to-sequence model, with 30k BPE operations, 6 encoder and decoder layers, 8 attention heads, 512 hidden units and 2048-dimensional feed-forward layers. Other hyperparameters were set to Sockeye's default.

The performance of our models on the WMT'17 test set are reported in the bottom of Table 1 under "NMT", together with the performance of Google translate. Based on Google translate's better performance for En→Ru, we used it for the evaluation.

3.2 Entity Mention Detection and Coreference Resolution

Entity Mention Detection and Coreference Resolution were performed sequentially, both with the goal of abstracting the entities present in the semantics of the document from the mentions, or *referents* present in the text. The underlying model is based on a coarse-to-fine model (Lee et al., 2018), which takes a candidate set of spans from within the text and clusters them based on mentions. For each span, a *span representation* is formed by pooling over the word embeddings of each word within the span. A mention scorer subsequently prunes all spans, leaving only likely referents. A coarse, similarity-based scorer then determines for each possible referent its top 50 most likely candidates antecedents based. A fine, neural scorer then scores these 50. During decoding, chains of referents are formed, each chain representing one entity. The entire model is typically trained (and evaluated) end-to-end on OntoNotes 5.0 (Pradhan et al., 2013), and the average test F1 reported with ELMo (Peters et al., 2018) is 73.0.

One drawback of the coarse-to-fine model is that it only returns chains of size two or more. In SM-KBP, we are also interested in entities that are mentioned once. We can additionally extract the high-

[2] http://data.statmt.org/wmt17/translation-task/preprocessed/ru-en

scoring spans according to mention scorer that were not chained, as that should capture the remaining singleton chains. However, those predictions only have a 70.1 F1 with noun-like phrases (according to gold syntactic parses). We attempted multi-task training of the noun-like phrases (objective is to minimize the loss of the mention scorer) and of the coreference chains (with the coreference cluster objective) and arrived at a compromise at 88.3 F1 against noun-like phrases and 72.4 coreference F1. However, the superior option was to train two separate models (a mention scorer and a coarse-to-fine linker) and pipe the span predictions of the first into the second. This yields a 94.1 F1 against noun-like phrases and 72.8 coreference F1. Finally, we used static BERT (Devlin et al., 2018) embeddings as input (with mean-pooling across subtokens), which improved the coreference F1 to 73.8.

3.3 Event Trigger Detection

An event trigger is the word or phrase that expresses the occurrence of an event. In order to identify triggers, We employed a linear chain Conditional Random Field (CRF) as a sequence tagger for trigger identification. The potential function of the CRF is generated by passing BERT features through a feed-forward neural network.

The model is pre-trained on PredPatt output[3] and fine-tuned on the LAO dataset, obtaining 83.9 F1.

3.4 Typing Entities, Events, and Relations

Given the BERT encodings of entities (the span), the event (the event trigger) and the relation (3 encodings, consisting of the relation trigger and the two argument spans), we classify them into a hierarchical ontology.

In the AIDA entity / event / relation hierarchy, there are 3 levels: types, subtypes, and sub-subtypes. Given the vector representation of an object (either the representation of an entity, an event, or a relation) $\mathbf{x}$ and its corresponding types y_1, y_2, y_3 (matching to the 3 levels in the ontology respectively), we employ a multi-level ranking loss:

$$\sum_{l=1}^{3} \sum_{y'_l \in Y'_l} [\xi_l - F(\mathbf{x}, y_l) + F(\mathbf{x}, y'_l)]_+ \qquad (1)$$

where l is the level of the ontology ranging from 1 to 3; y'_l are negative samples on level l; $F(\mathbf{x}, y)$ is a scoring function between the object and the type; and ξ_l is a margin hyperparameter for level l.

Intuitively, coarser types are easier and finer types are harder for classifiers — hence larger margin hyperparameter for coarser types should yield better performance. Our preliminary experiments indeed show that this is the case — a graded set of margins perform better than a uniform set of margins. In our experiments, $(\xi_1, \xi_2, \xi_3) = (0.75, 0.5, 0.25)$.

[3] https://github.com/hltcoe/PredPatt

The typing scorer function $F(\mathbf{x}, y)$ follows the matching function in Mou et al. (2016): we first concatenate $\mathbf{x}$ and a type embedding for type y, together with their elementwise product and elementwise distance, then pass through a multi-layer feed-forward neural network with ReLU activation functions in between.

We randomly sample 80% samples for training and 20% as development in the AIDA practice annotation dataset. For the event typer, on English, we get 0.741, 0.721 and 0.675 accuracy for the 3 levels. For the entity typer, 0.641, 0.438 and 0.419; the relation typer, 0.430, 0.418 and 0.413.

3.5 Entity Linking: Textual Similarity Matching

We follow the approach we used in the AIDA M9 evaluation. Namely, we determine a match between a document mention and an entity string from a hypothesis by a string similarity model (Neculoiu et al., 2016). Given two strings, the model outputs a score indicating how similar these two input strings are. The model is a Siamese network where two identical, parameter-shared modules are stacked upon the two input strings, with each string considered as a character sequence. Each module comprises of 4 bidirectional LSTM layers (size 64) followed by a feedforward layer that results in a vector of size 128. The model was trained using a dataset based on WikiNames, arising from earlier research at JHU (Andrews et al., 2012).

3.6 Argument Linking

Argument linking is the task of determining which mentions are arguments for the events evoked by the event triggers and which role each argument serves. We perform this task in two steps for a given event: 1) for each role, we predict mentions that best fill the role (with confidences); 2) we jointly resolve the trigger's type and each mention's type (§3.4) by greedily selecting the types that yield the highest overall confidence (link confidence and type confidence) while also obeying the type constraints in the LAO.[4]

We adapt the semantic role labeling model proposed by (He et al., 2018), modified for the AIDA task. The original neural model learns span representations for predicates (triggers) and arguments (mentions), then scores them jointly with role-specific parameters to choose the best role for a predicate-argument pair. In our model, we instead find the best mention that fills a given role for a given trigger. We additionally extend the model to operate across sentences using techniques from (Lee et al., 2017, 2018), as many of the arguments in the training annotations do not appear in the same sentence as their trigger (23.1% of event arguments are out-of-sentence, 5.5% for relation arguments, 16.1%

[4] If a mention does not obey the LAO type constraints, we consider the mention that fills the role with the next most confidence.

overall).[5] We note that restricting the model to operate on only in-sentence arguments may have led in the evaluation to higher performance, trading off recall for precision, but we operated on the assumption that we should employ models that could concievably capture any of the arguments that were being annotated by the LDC.

We pretrained the argument linking model with our new crowd-sourced data, then finetuned on the practice annotations augmented with event and entity mentions predicted by upstream components (to match test-time conditions). All training data was in English. Because the argument linking model does not observe or predict type information (types are resolved in post-processing), we report performance on untyped (trigger, role, mention) links. On development data, our model achieved precision of 43% and recall of 58%, yielding F_1 of 49.6% for untyped link prediction.

3.7 Alignment

The evaluation scenario requires extracting information in the source language. Since we first translated Ukrainian and Russian into English to run target-side information extraction tools, we need a way to map spans extracted in English back to spans in the source language. In addition, when receiving a query span for a Ukrainian or Russian document, we need to map this span onto its English translation.

Both these problems can be solved with alignment models. We trained alignment models in all 4 possible directions (Uk $\leftrightarrow$ En, Ru $\leftrightarrow$ En) following Stengel-Eskin et al. (2019). Due to a lack of gold alignments for Russian and Ukrainian, we used `fast-align` (Dyer et al., 2013) on the datasets used for Machine Translation (see Section 3.1).

References

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 344–355.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Ulrich Germann, Alexandra Birch, Faheem Kirefu, Peter Bell, Hervé Bourlard, Steve Renals, Sebastião Miranda, David Nogueira, Simon Vandieken, Andrea Carmentini, and Ahmed Ali. 2019. Scalable understanding of multilingual media (summa).

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, August 11, 2016*, pages 148–157.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

[5] These numbers are only for arguments that are entity mentions. About 12.5% of all relation arguments are event mentions (events do not take other events as arguments under the LAO).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Elias Stengel-Eskin, Tzu-Ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. *arXiv preprint arXiv:1909.00444*.