GAIA at SM-KBP 2020 - A Dockerlized Multi-media Multi-lingual Knowledge Extraction, Clustering, Temporal Tracking and Hypothesis Generation System

Manling Li[1], **Ying Lin**[1], **Tuan Manh Lai**[1], **Xiaoman Pan**[1], **Haoyang Wen**[1], **Sha Li**[1]
Zhenhailong Wang[1], **Pengfei Yu**[1], **Lifu Huang**[1], **Di Lu**[1], **Qingyun Wang**[1]
Haoran Zhang[1], **Qi Zeng**[1], **Chi Han**[1], **Zixuan Zhang**[1], **Yujia Qin**[1]
Xiaodan Hu[1], **Nikolaus Parulian**[1], **Daniel Campos**[1], **Heng Ji**[1]
[1] University of Illinois at Urbana-Champaign hengji@illinois.edu
Brian Chen[2], **Xudong Lin**[2], **Alireza Zareian**[2], **Amith Ananthram**[2], **Emily Allaway**[2]
Shih-Fu Chang[2], **Kathleen McKeown**[2]
[2] Columbia University sc250@columbia.edu, kathy@cs.columbia.edu
Yixiang Yao[3], **Michael Spector**[3], **Mitchell DeHaven**[3],
Daniel Napierski[3], **Marjorie Freedman**[3], **Pedro Szekely**[3]
[3] Information Sciences Institute, University of Southern California mrf@isi.edu
Haidong Zhu[4], **Ram Nevatia**[4]
[4]University of Southern California nevatia@usc.edu
Yang Bai[5], **Yifan Wang**[5], **Ali Sadeghian**[5], **Haodi Ma**[5], **Daisy Zhe Wang**[5]
[5] University of Florida daisyw@ufl.edu

1 Introduction

We participated in the SM-KBP 2020 evaluation using our dockerlized GAIA system, an end-to-end knowledge extraction, grounding, inference, clustering, temporal tracking and hypothesis generation system, as shown in Figure 1. Our TA1 system achieves top performance at both intrinsic evaluation and extrinsic evaluation through TA2 and TA3. In the past year, we integrate the following innovations:

- **Multilingual Joint Information Extraction with Global Knowledge:** We propose an end-to-end neural model ONEIE to extract entities, relations and events jointly in a language independent fashion. Existing joint neural models for Information Extraction (IE) use local task-specific classifiers to predict labels for individual instances (e.g., trigger, relation) regardless of their interactions. For example, a VICTIM of a DIE event is likely to be a VICTIM of an ATTACK event in the same sentence. Our model can capture such cross-subtask and cross-instance inter-dependencies, we extract the globally optimal information network by considering the inter-dependency among nodes and edges. At the decoding stage, we incorporate global features to capture the cross-subtask and cross-instance interactions. As ONEIE does not use any language-specific feature, we prove it can be easily applied to new languages or trained in a multilingual manner.

- **Document-Level Event Argument Role Labeling:** Event extraction has long been treated as a sentence-level task in the Information Extraction community. We argue that this setting does not match human informative seeking behavior and leads to incomplete and uninformative extraction results. We propose a document-level neural event argument extraction model by formulating the task as conditional generation following event templates.

- **Symbolic Semantics Enhanced Event Coreference Resolution:** We propose a novel context-dependent gated module to incorporate a wide range of symbolic features (e.g., event types and attributes) into event coreference resolution. Simply concatenating symbolic features with contextual embeddings is not optimal, since the features can be noisy and contain errors. Also, depending on the context, some features can be more informative than others. Therefore, the gated module extracts information from the symbolic features selectively. Combined

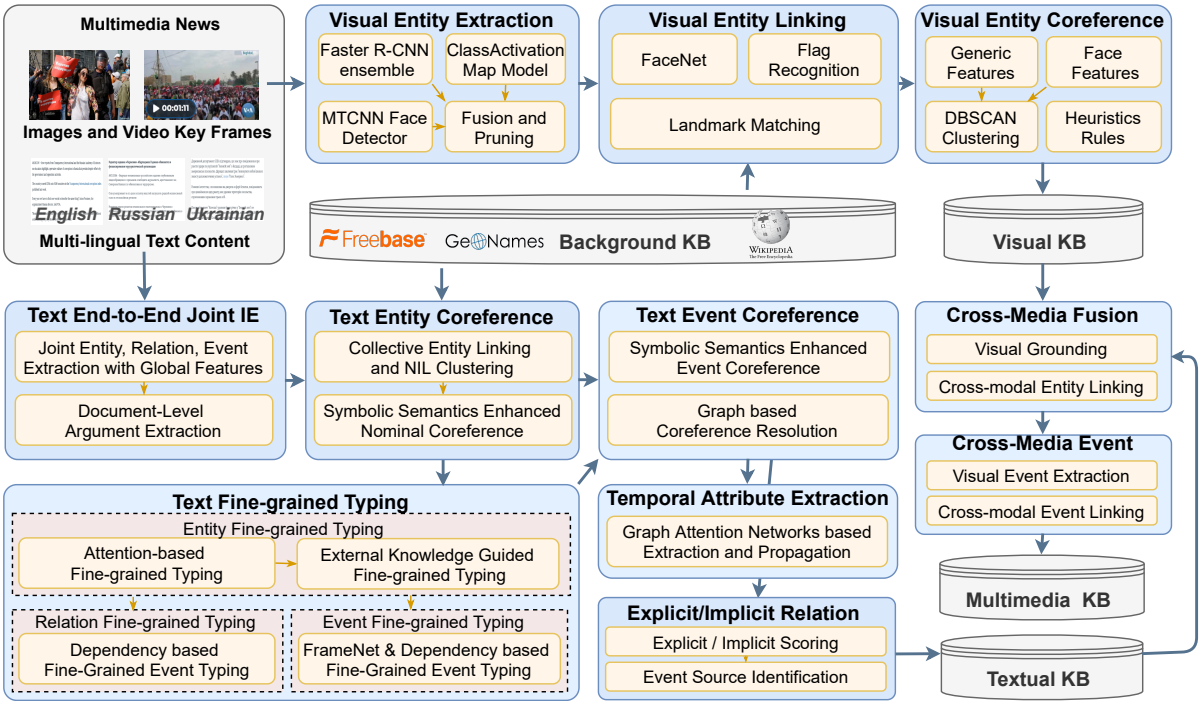

Figure 1: The architecture of GAIA multimedia knowledge extraction system.

with a simple regularization method that randomly adds noise to the features during training, our best event coreference models achieve state-of-the-art results on public benchmark datasets such as ACE 2005 and KBP 2016.

- **Event Temporal Attribute Extraction and Propagation via Graph Attention Networks:** We propose a graph attention networks based approach to propagate temporal information over document-level event graphs constructed by shared entity arguments and temporal relations. To better evaluate our approach, we have developed a challenging new benchmark, where more than 78% of events do not have time spans mentioned explicitly in their local contexts. The proposed approach yields an absolute gain of 7.0% in match rate over contextualized embedding approaches, and 16.3% higher match rate compared to sentence-level manual event time argument annotation.

- **Implicit/Explicit Relation Extraction and Source Identification:** We extend our information extraction capabilities with an ensemble of neural zero-shot and few-shot techniques designed to identify a subset of relation types whose expression is both explicit and implicit (like *blame*). In addition to these challenging relation types, this component also identifies source information for every event, enabling better perspective clustering during TA3 hypothesis generation.

- **Cross-media Structured Common Semantic Space for Multimedia Event Extraction:** We propose and develop a new multimedia Event Extraction (M2E2) task that involves jointly extracting events and arguments from text and image. We propose a weakly supervised framework which learns to encode structures extracted from text and images into a common semantic embedding space. This structured common space enables us to share and transfer resources across data modalities for event extraction and argument role labeling.

- **Video Multimedia Event Extraction and argument labeling** We extend the multimedia Event Extraction (M2E2) task to extracts events and arguments from videos and article pairs. We propose a self-supervised multimodal transformer that learns the multimodal context from each modality by utilizing the self-attention mechanism and learning

to predict the event types and argument roles from both modalities in a sequential decoder. This proposed architecture allows us to fully learn the interaction between event and argument information from both modalities and jointly extract events and argument roles.

2 TA1 Text Knowledge Extraction

2.1 Approach Overview

We dockerize an end-to-end fine-grained knowledge extraction system for 179 entity types, 149 event types, and 50 event types defined in AIDA ontology. As shown in Figure 1, it supports the joint extraction of entities, relations and events from multilingual corpus (English, Russian and Spanish), and performs coreference resolution over entities and events. We will present the details of each component in the following sections.

2.2 Joint Entity, Relation and Event Mention Extraction

We use a sentence-level joint neural model (Lin et al., 2020) to extract entities, relations, and events from text. For English, we train two separate IE models. The first model is trained on ACE and ERE English data that are mapped to the AIDA ontology. Another model is trained on documents we annotate for new AIDA types. Similarly, we trained two IE models for Spanish on ERE data and our own annotations respectively. We further enhance the Spanish model with transfer learning by adding English training data with a lower sampling rate (0.1 in our experiments). For Russian, we only train a single model on our Russian and English annotations in a multilingual way because it is not included in ACE or ERE. We use RoBERTa (Liu et al., 2019) for English and XLM-RoBERTa (Conneau et al., 2019) for Spanish and Russian to obtain contextualized word representations.

2.3 Document-level Argument Extraction

Given the event triggers detected from the previous module, document-level argument extraction aims to additionally look for cross-sentence arguments.

We frame this problem as conditional generation given an event template. An example of the converted input and output is shown in Figure 2. The event template is a sentence that describes arguments of an event type with $\langle arg \rangle$ placeholders.

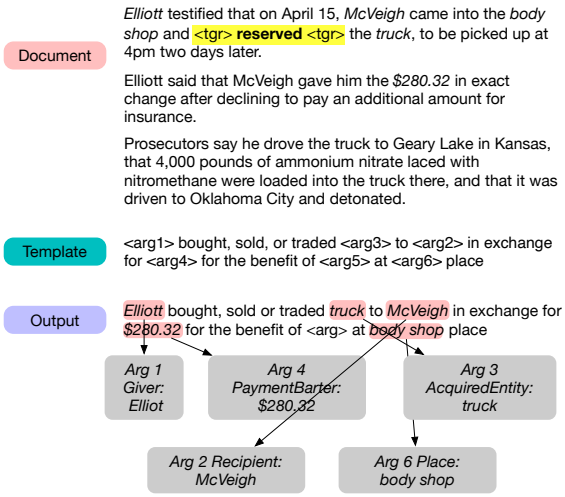

Figure 2: An example of document-level argument extraction formulated as text generation.

The generated output is a filled template where placeholders are replaced by concrete arguments. Note that one template is used for all event instances within the same type and such templates are already part of the AIDA ontology.

Our base model is an encoder-decoder language model BART (Lewis et al., 2020). The generation process models the conditional probability of selecting a new token given the previous tokens and the input to the encoder. To utilize the encoder-decoder LM for argument extraction, we construct an input sequence of $\langle$s$\rangle$ template $\langle s \rangle \langle / s \rangle$document $\langle / s \rangle$. All argument names (arg1, arg2 etc.) in the template are replaced by a single special placeholder token $\langle$arg$\rangle$.

The generation probability is computed by taking the dot product between the decoder output and the embeddings of tokens from the input. To prevent the model from hallucinating arguments, we restrict the vocabulary of words to V_c: the set of tokens in the input.

$$\mathrm{p}(\mathrm{x}_i = w | x_{<i}, c, t) = \begin{cases} \text{Softmax}\left(h_i^T \text{Emb}(w)\right) & w \in V_c \\ 0 & w \notin V_c \end{cases} \quad (1)$$

The model is trained end-to-end by minimizing the negative loglikelihood over all (content, template, output) instances in the dataset D:

$$\mathcal{L}(D) = -\sum_{i=1}^{|D|} \log p_\theta \left(x^i \mid c^i \right) \quad (2)$$

Missing arguments and multiple arguments in the same role are common cases in the argument extraction task. For missing arguments, we train the model to generate the special $\langle$arg$\rangle$ token. For

multiple arguments, we add the keyword "add" between the arguments. For example in ACE 2005 we have this sentence: "Afterwards Shalom was to fly on to London for talks with British Prime Minister Tony Blair and Foreign Secretary Jack Straw.". The input template is "⟨arg⟩ met with ⟨arg⟩ at ⟨arg⟩ place" and the generation output is "Shalom met with Tony Blair and Jack Straw at London place".

To align the predictions of the model back to the text for downstream modules, we adopt the simple heuristic of matching the closest occurrence of the predicted argument to the trigger.

2.4 Informative Justification Extraction

For named entities, we generate informative justification using the longest name mention. For nominal entities, we apply a syntactic tree parser[1] and select the sub-tree whose syntactic head word matches the nominal entity mention. For events, we use the first substring covering the trigger word and arguments as informative justification.

2.5 Fine-grained Typing

We follow (Li et al., 2019) to detect fine-grained types for entities, relations and events. For event fine-grained typing, we annotate the newly added ten event types and train an extractor for these new types.

As aforementioned in Section 2.2, we train separate IE models on different datasets and combine their outputs. Although ACE and ERE datasets contain much more training instances with higher annotation quality, they only cover an incomplete set of event types in the AIDA ontology. By contrast, our new datasets are smaller but have a more complete coverage of the new types. Therefore, we prioritize results predicted by models trained on ACE and ERE data when resolving conflicts in the process of merging IE results. For example, if the first model predicts "Brooklyn Bridge" as a FAC entity, while the second model predicts it as a LOC, we keep the FAC label in this case.

2.6 Entity Linking and Coreference Resolution

2.6.1 Entity Linking

We follow (Li et al., 2019) to link entities to background KB and Freebase for English and Russian. For Spanish, we use translation dictionaries mined from Wikipedia (Ji et al., 2009) to translate each mention into English first.

2.6.2 Entity Coreference Resolution

For Russian entity coreference resolution, we follow the approach of Li et al. (2019). For English and Spanish, we implement neural models similar to the `bert-coref` model (Joshi et al., 2019). However, there are several important differences. First, we remove the higher-order inference (HOI) layer (Lee et al., 2018) from the original architecture. Our preliminary results suggest that HOI typically does not improve the coreference resolution performance while incurring additional computational complexity. This observation agrees with a recent analysis of Xu and Choi (2020). Second, we also apply a simple heuristic rule based on the entity linking results to refine the predictions of the neural models. We prevent two entity mentions from being merged together if they are linked to different entities with high confidence. For English, we use SpanBERT (large) (Joshi et al., 2020) as the Transformer encoder and train the system on ACE 2005 (Walker et al., 2006), EDL 2016[2], EDL 2017[3], and OntoNotes (English) (Pradhan et al., 2012). For Spanish, we use XLM-Roberta (large) (Conneau et al., 2020) and train the system on OntoNotes (Spanish) (Pradhan et al., 2012), DCEP (Dias, 2016), and SemEval 2010 (Recasens et al., 2010).

2.7 Event Coreference Resolution

For Russian event coreference resolution, we follow the approach of Li et al. (2019). For English and Spanish, we implement a single cross-lingual model that incorporates a wide range of symbolic features into event coreference resolution. Given an input document D consisting of n tokens, our model first forms a contextualized representation for each input token, using the multilingual XLM-RoBERTa (XLM-R) Transformer model (Conneau et al., 2020). Let $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ be the output of the Transformer encoder, where $\mathbf{x}_i \in \mathbb{R}^d$.

Single-Mention Encoder For each (predicted) event mention m_i, its trigger's representation $\mathbf{t}_i$ is defined as the average of its token embeddings:

$$\mathbf{t}_i = \sum_{j=s_i}^{e_i} \frac{\mathbf{x}_j}{e_i - s_i + 1} \quad (3)$$

[1] https://spacy.io/

[2] LDC2017E03

[3] LDC2017E52

We assume that each m_i has K different symbolic features associated with it (e.g., its predicted event type and attributes). Using K trainable embedding matrices, we convert the symbolic features of m_i into K vectors $\{\mathbf{h}_i^{(1)}, \mathbf{h}_i^{(2)}, \ldots, \mathbf{h}_i^{(K)}\}$, where $\mathbf{h}_i^{(u)} \in \mathbb{R}^l$.

Mention-Pair Encoder Given two event mentions m_i and m_j, we define their trigger-based pair representation as:

$$\mathbf{t}_{ij} = \text{FFNN}_\text{t}\big(\big[\mathbf{t}_i, \mathbf{t}_j, \mathbf{t}_i \circ \mathbf{t}_j\big]\big) \tag{4}$$

where FFNN_t is a feedforward network mapping from $\mathbb{R}^{3\times d} \rightarrow \mathbb{R}^p$, and $\circ$ is element-wise multiplication. Similarly, we compute their feature-based pair representations $\{\mathbf{h}_{ij}^{(1)}, \mathbf{h}_{ij}^{(2)}, \ldots, \mathbf{h}_{ij}^{(K)}\}$ as follows:

$$\mathbf{h}_{ij}^{(u)} = \text{FFNN}_\text{u}\big(\big[\mathbf{h}_i^{(u)}, \mathbf{h}_j^{(u)}, \mathbf{h}_i^{(u)} \circ \mathbf{h}_j^{(u)}\big]\big) \tag{5}$$

where $u \in \{1, 2, \ldots, K\}$, and FFNN_u is a feedforward network mapping from $\mathbb{R}^{3\times l} \rightarrow \mathbb{R}^p$.

Symbolic Features Incorporation In our dockerlized GAIA system, we predict the symbolic features using simple predictors. As a result, the symbolic features can be noisy and contain errors. Also, depending on the specific context, some features can be more useful than others. Inspired by previous studies on gating mechanisms (Lin et al., 2019; Lai et al., 2019), we propose **Context-Dependent Gated Module** (CDGM), which uses a gating mechanism to extract information from the input symbolic features selectively. Given two mentions m_i and m_j, we use their trigger feature vector $\mathbf{t}_{ij}$ as the main controlling context to compute the filtered representation $\overline{\mathbf{h}}_{ij}^{(u)}$:

$$\overline{\mathbf{h}}_{ij}^{(u)} = \text{CDGM}^{(u)}\big(\mathbf{t}_{ij}, \mathbf{h}_{ij}^{(u)}\big) \tag{6}$$

where $u \in \{1, 2, \ldots, K\}$. More specifically:

$$\begin{aligned}
\mathbf{g}_{ij}^{(u)} &= \sigma\big(\text{FFNN}_\text{g}^{(u)}\big(\big[\mathbf{t}_{ij}, \mathbf{h}_{ij}^{(u)}\big]\big)\big) \\
\mathbf{o}_{ij}^{(u)}, \mathbf{p}_{ij}^{(u)} &= \text{DECOMPOSE}\big(\mathbf{t}_{ij}, \mathbf{h}_{ij}^{(u)}\big) \\
\overline{\mathbf{h}}_{ij}^{(u)} &= \mathbf{g}_{ij}^{(u)} \circ \mathbf{o}_{ij}^{(u)} + \big(1 - \mathbf{g}_{ij}^{(u)}\big) \circ \mathbf{p}_{ij}^{(u)}
\end{aligned} \tag{7}$$

where σ denotes sigmoid function. $\text{FFNN}_\text{g}^{(u)}$ is a mapping from $\mathbb{R}^{2\times p} \rightarrow \mathbb{R}^p$. At a high level, $\mathbf{h}_{ij}^{(u)}$ is decomposed into an orthogonal component and a parallel component, and $\overline{\mathbf{h}}_{ij}^{(u)}$ is simply the fusion of these two components. In order to find the optimal mixture, $\mathbf{g}_{ij}$ is used to control the composition. The decomposition unit is defined as:

$$\begin{aligned}
&\textit{Parallel} \quad \mathbf{p}_{ij}^{(u)} = \frac{\mathbf{h}_{ij}^{(u)} \cdot \mathbf{t}_{ij}}{\mathbf{t}_{ij} \cdot \mathbf{t}_{ij}} \mathbf{t}_{ij} \\
&\textit{Orthogonal} \quad \mathbf{o}_{ij}^{(u)} = \mathbf{h}_{ij}^{(u)} - \mathbf{p}_{ij}^{(u)}
\end{aligned} \tag{8}$$

where $\cdot$ denotes dot product. The parallel component $\mathbf{p}_{ij}^{(u)}$ is the projection of $\mathbf{h}_{ij}^{(u)}$ on $\mathbf{t}_{ij}$. It can be viewed as containing information that is already part of $\mathbf{t}_{ij}$. $\mathbf{o}_{ij}^{(u)}$ is orthogonal to $\mathbf{t}_{ij}$, and so it can be viewed as containing *new* information.

Mention-Pair Scorer After using CDGMs to distill symbolic features, the final pair representation $\mathbf{f}_{ij}$ of m_i and m_j can be computed as follows:

$$\mathbf{f}_{ij} = [\mathbf{t}_{ij}, \overline{\mathbf{h}}_{ij}^{(1)}, \overline{\mathbf{h}}_{ij}^{(2)}, \ldots, \overline{\mathbf{h}}_{ij}^{(K)}] \tag{9}$$

And the coreference score $s(i, j)$ of m_i and m_j is:

$$s(i, j) = \text{FFNN}_\text{a}(\mathbf{f}_{ij}) \tag{10}$$

where FFNN_a is a mapping from $\mathbb{R}^{(K+1)\times p} \rightarrow \mathbb{R}$.

Noisy Training We use the same loss function as in (Lee et al., 2017). We also notice that the training accuracy of a feature predictor is typically near perfect. Therefore, if we simply train our model without any regularization, our CDGMs will rarely come across noisy symbolic features during training. Therefore, to encourage our CDGMs to actually learn to distill reliable signals, we also propose a simple but effective **noisy training** method. Before passing a training data batch to the model, we randomly add noise to the predicted features. More specifically, for each document D in the batch, we go through every symbolic feature of every event mention in D and consider sampling a new value for the feature.

Training Datasets For English, we train the system on ACE 2005 (Walker et al., 2006) and KBP 2016 (Mitamura et al., 2016). For Spanish, we train the system on ERE-ES (Song et al., 2015).

2.8 Temporal Attribute Extraction

For English documents, we first use Stanford CoreNLP (Manning et al., 2014) to perform time expression extraction and normalization for all documents. Then we perform sentence-level time argument extraction. Specifically, we fine-tuned BERT on ACE 2005 event time argument annotations. We use the representation of the first token

of an event span and a time span to perform pairwise classification.

We further propagate local event time to document-level using graph attention networks (Velickovic et al., 2018). We construct document-level event graphs as $G = \{(e_i, v_j, r_{i,j})\}$, where each bi-directed edge $r_{i,j}$ represents the argument role between an event e_i and an entity or time expression v_j. We first obtain token representation from BERT for all sentences in a document. Then we use the average representation for event triggers, entities and time expressions that contains multiple tokens. To propagate information from connected nodes, we use a two-layer graph attention networks that will update the representations for events, entities and time expressions. We use a two-layer feed-forward networks to estimate the probability to fill time expression t_j in event e_i's 4-tuple time elements. To resolve conflict, we use a greedy approach that choose 4-tuple element candidates based on the descending order of their probabilities, and fill in the time if there is no conflict, otherwise we drop the candidate.

For English relations, Spanish and Russian events and relations, we use the document creation time as the latest start time and earliest end time.

3 TA1 Explicit/Implicit Relation Extraction

We employ a separate component to handle the extraction of relations in the AIDA ontology whose expression is more diverse than standard ontological relations like $father - of$. These relations are sponsorship, blame, deliberateness, legitimacy, hoax-fraud, and sentiment. Extracting these types is extremely challenging as 1) they are data scarce (there are few, if any, gold label examples) and 2) they can be expressed both explicitly, using identifiable trigger words, and implicitly. For example, the blame relation is clear in both "Maduro blamed the protestors for the attack and "Maduro had the protestors arrested for the attack" but in the latter it must be inferred. As such, we deploy an ensemble of few-shot techniques for explicit and implicit information extraction.

3.1 Explicit Relation Scoring

To extract explicit relations, we incorporate our work on few-shot neural relation extraction (Ananthram et al., 2020). It builds on the current start-of-the-art, "Matching the Blanks" (MTB) (Soares et al., 2019) which extends Harris' distributional hypothesis (Harris, 1954) to relations. Soares et al. assume that the informational redundancy of very large text corpora (e.g., Wikipedia) results in sentences that contain the same pair of entities generally expressing the same relation. Thus, an encoder trained to collocate such sentences can be used to identify the relation between entities in any sentence s by finding the labeled relation example whose embedding is closest to s.

While MTB is very successful, it relies on a huge amount of data, making it difficult to retrain in English or any other language with standard computational resources. To address this challenge, we assume that sections of news corpora exhibit even more informational redundancy than Wikipedia. Specifically, news in the days following an event (e.g., the 2006 World Cup) frequently re-summarizes the event before adding new details. As a result, news exhibits a strong form of local consistency over short rolling time windows where otherwise fluid relations between entities remain fixed. For example, the relation between Italy and France as expressed in a random piece of text is dynamic and context-dependent, spanning a wide range of possibilities that include "enemies", "neighbors" and "allies". But, in the news coverage following the 2006 World Cup, it is static – they are sporting competitors. Therefore, by considering only sentences around specific events, we extract groups of statements that express the same relation and are relatively free of noise.

Using this method, we extract a distantly supervised training corpus in English, Spanish and Russian from the Reuters RCV1 and RCV2 newswire corpora (Lewis et al., 2004) guided by date-marked event descriptions from Wikipedia. We use this corpus to train multilingual BERT (Devlin et al., 2018) to produce high quality general-purpose relation representations from relation statements. We adopt the common definition of a relation statement in the literature: a triple $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$ where $\mathbf{x} = [x_0 \ldots x_n]$ is a sequence of tokens and $\mathbf{s}_1 = (i, j)$ and $\mathbf{s}_2 = (k, l)$ are the indices of special start and end identifier tokens that demarcate the two entity mentions in $\mathbf{x}$. mBERT maps this relation statement to a fixed-length vector $h \in \mathbb{R}^d$. The vector h represents the relation between the entity mentions identified by $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ as expressed in $\boldsymbol{x}$. The cosine similarity be-

tween $f(\boldsymbol{r})$ and $f(\boldsymbol{r}_O)$ should be close to 1 if and only if $\boldsymbol{r}$ and $\boldsymbol{r}_O$ express the same relation. That is to say, mBERT should collocate sentences that exhibit similar relations.

To incorporate this work into the AIDA pipeline, we rely on the entity and event extractions from earlier components to produce candidate relation statements for the AIDA corpus. We compare each candidate to the gold labeled exemplars for each relation provided by LDC, producing similarity scores for each candidate / relation exemplar pair between 0 and 1. These scores are then considered by our final aggregation step when deciding whether or not to accept a particular candidate relation statement.

3.2 Implicit Relation Scoring

To identify implicit relations, we augment relation candidates with stance (pro, con and neutral) scores meant to capture the valence towards a particular entity or event whose expression may be subtle. This information provides useful signal for the identification of relations that have intrinsic positive or negative connotations. For example, sentences that blame an individual for an event often take a negative position towards that individual that can only be inferred implicitly (e.g., "Maduro blamed outside *agitators* for the attack").

To produce these scores, we incorporate our work on zero-shot stance detection (Allaway and McKeown, 2020). In that work, we present a new dataset for the challenging task of generalizable stance detection on unseen topics. This corpus captures a wider range of topics and lexical variation than in previous datasets. Using this dataset, we design and train a new model for stance detection that captures relationships between topics without supervision and beats the state-of-the-art on a number of challenging linguistic phenomena.

This new model, Topic-Grouped Attention (TGA) Net, consists of 1) a BERT-based contextual conditional encoding layer, 2) topic-grouped attention using generalized topics representations and 3) a feed-forward neural network (see Figure 3). Given a sentence s and a topic t, the contextual conditional encoding layer first embeds the pair using BERT (Devlin et al., 2018), resulting in sequences of token embeddings for the sentence s and for the topic t. We use a concatenation of tf-idf weighted averages of the embeddings of s and t to find the closest cluster in a hierarchical clustering of sentences and topics from our training data and treat its centroid as our generalized topic representation r. Using r, we compute the similarity between t and all topics seen during training via learned scaled dot-product attention (Vaswani et al., 2017) and use these similarity scores to produce a weighted average of our topic tokens c that captures the relationship between t and related topics and documents. Finally, we concatenate our embeddings of s and t with c and pass it through several feed-forward layers to produce a probability distribution over our three stance labels: pro, con and neutral.

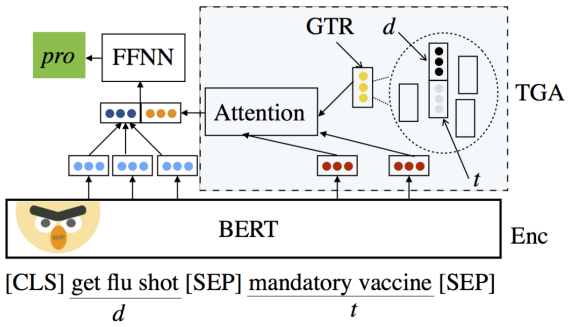

Figure 3: Architecture of TGA Net. Enc indicates contextual conditional encoding, GTR indicates Generalized Topics Representation, TGA indicates Topic-grouped Attention

To incorporate this work into the AIDA pipeline, we augment every relation candidate with the stance score towards each entity or event in the relation statement. As with our explicit relation scores, these scores are considered by our final aggregation step when deciding whether or not to accept a particular candidate relation statement.

3.3 Aggregating Scores

In addition to our new explicit and implicit relation scoring components, we augment our candidate relation statements with trigger-based and sentiment-based scores from our existing system, presented as part of (Li et al., 2019). We use highly regularized decision trees trained on dozens of examples from AIDA practice corpora which we manually annotated to make the ultimate acceptance decision based on these scores.

3.4 Event Source Information

Finally, to enable better perspective clustering during TA3 hypothesis generation, we adapt our explicit relation extraction system to identify the

source of all event information along with a confidence score. For example, in the sentence "Maduro says the protests seeking to oust him are backed by the United States.", we identify "Maduro" as the source of extracted Protest event.

4 TA1 Visual Knowledge Extraction

We first review our Visual Knowledge Extraction (VKE) system (Li et al., 2019) last year and introduce our new component from our current system. Our system further combines information from multimodal sources at the entity level (grounding) and at the event level (event-type, argument roles), which serves multimodal information from different modalities as complementary to each other.

4.1 Entity Detection

The object detection system contains four different systems: three Faster R-CNN (Ren et al., 2015) models and a weakly supervised CAM model (Zhou et al., 2016). We followed the same process (Li et al., 2019) to aggregate the results from a different model and created a new mapping for the classes to the new m36 ontology. For face detection, we use an MTCNN model (Zhang et al., 2016). For the overlapped detection between the general object model and face model, we create a cluster using the object detection result with the largest bounding box as the prototype to represent the detected result.

4.2 Entity Recognition

The entity recognition pipeline is done by face recognition models FaceNet (Schroff et al., 2015) where we recognize predefined name list recognized by the text named-entity recognition model. We covered around 500 names in our current system.

4.3 Cross-Modal Entity Coreference

The entity coreference pipeline aims to build a knowledge graph by linking detected entities by our entity detection component. Our entity coreference model has two components: single-modality and cross-modality. The single-modality component finds entities that co-occur in multiple images within the same root document. The cross-modality component links the entity extracted from the text model to the entities in the images. This year, the cross-modal coreference model links entity-level information (object from images and entities from the text) and is used to discover event-level information.

We followed our previous visual grounding system (Zhang et al., 2018) which extracts a multi-level visual feature map for each image in a document. For each word (or phrase, entity mention, etc.), we compute an attention map to every feature map location to localize the query by computing the similarity between the word and region. On the other hand, our network takes each sentence of the document and represents each word using a language model. We calculate the sentence to image similarity score using all pairs in the document to find potential co-referenced events across modality. Details will be described in the later section.

4.4 Event and argument role extraction

Besides extracting entity information from images and videos, we also extract visual events and their argument roles from visual data. To train our system, we have collected a dataset called video M2E2, which contains 4.5K video-article pairs from YouTube news channels. We start from 20 event types defined in AIDA ontology, which is visually detectable and search on news channels. In the end, we annotated 1.2K video article pairs for training and evaluation. Given the annotation, we have developed several models on top of this data. First, we have trained an image-based model using Joint Situation Localizer (JSL (Pratt et al., 2020)). We combine the annotation of video M2E2 and the Swig (Pratt et al., 2020) data and map the event types and argument roles to the AIDA ontology. In this setting, the model can detect argument roles that were not defined in the Swig data, such as visual display in the protest event.

4.5 Multimodal event coreference

We further extended this model to find event coreference between image and text events. For the images with detected events, we apply our previous grounding model to find sentences within the same root document with high image-sentence similarity, representing the sentence content similar to the image content. Also, we find the event mention in the sentence extracted by the text event extraction tool. We apply a rule-based approach to determine if the image event and the event mention in the sentence have a coreference relation. (1) The event type of the event mention in the sentence has the same event type extracted in the image.

(2) The image and sentence have a high similarity score. (3) No contradiction in the entity types for the same argument role across different modalities. If all three criteria are valid, we determine that the two events from different modalities have a coreference relation. This pipeline allows us to find 36% of visual events contain additional arguments not mentioned in the text, with 98 additional arguments detected. For the event detection performance, visual events had a precision of 60%, and visual events with coreference had a precision of 82%. So the step of co-referencing with text events serves as a useful filtering step to further enhance visual event detection accuracy.

5 TA2 Cross-Document Coreference

Our TA2 focuses on generating high precision clusters of entities across documents since the incoming data includes noisy extractions and has missing information. For the named entity, each entity contains limited labels and pre-linked external knowledge base identifiers with confidence. The simple but effective clustering algorithm maps all entities with identifiers to Wikidata and initializes clusters with knowledge base identifiers. The labels of these clusters get enriched from Wikidata's multilingual label, aliases and descriptions. Each cluster then computes several trusted labels for attracting other entities without knowledge base identifiers and these newly merged entities must have compatible types with the cluster type. For the rest of the entities, they form singleton clusters and get merged based on the similarity of labels. Finally, a prototype is elected from all entities within each cluster to represent the whole cluster based on its extraction confidence and label prevalence in the cluster. To deal with the large input triples in AIDA Interchangeable Format, we uses KGTK [4] which is a flexible and low-resource required python library for knowledge graph manipulation in TSV intermediate format.

6 HypoGator: Alternative Hypotheses Generation and Ranking

HypoGator is the hypothesis Generation system developed by the University of Florida. With a search-rank-cluster approach, it finds alternative perspectives to complex topics(queries) over the automatically extracted knowledge graph by TA1 and TA2. Briefly speaking, HypoGator decomposes a complex graph query into subqueries of simple subgraph patterns. For each subquery its entry points are matched into the inferred input knowledge graph and their local context generates candidate answers. Candidates are scored and ranked using multiple features that are indicative of coherence and relevance. A join algorithm combines the answers from each atomic query and re-scores the final set of answers using features that encourage answer cohesion. Finally, a newly developed hypotheses clustering algorithm is applied to select out the alternative hypotheses based on both structural and semantic features. Figure 4. Details of the core components are covered in the following subsections.

6.1 Query Processing

A statement of information need(SIN) is a subgraph pattern with event/relation types and entities as nodes, event/relation argument roles as edges and a set of grounded entities known as entry points. We classify an SIN as simple if each entry points is used as the argument of only one event/relation. In contrast, a complex SIN has entry points that are shared by multiple events/relations developing into a star-like structure. HypoGator's query processing module first scan an SIN and decomposes a complex SIN into multiple simple SINs that we refer to as atomic. The decomposition algorithm first finds all connected components in the SIN and for each component, the algorithm visits the neighbors of its entry points and traverse each of them until a different entry point or a terminal node is found. The resulting subgraphs are added to the atomic query list. Figure 5 shows an example SIN with entry points *Odessa (a.k.a. Odesa)* and *Trade Unions House (a.k.a. Trade Unions Building)* in the center and atomic queries derived from it using the decomposition algorithm round it.

After query decomposition, HypoGator matches entry points into the inferred knowledge graph. Since it is common to see that the information of entities in the given KG are incomplete, HypoGator will try to use all the available entry point information given by the SIN for matching separately, including the background KB id, the provenance offset and the strings of all the names/alias of an entry point. When doing string matching, common string similarity metrics and

[4] https://github.com/usc-isi-i2/kgtk

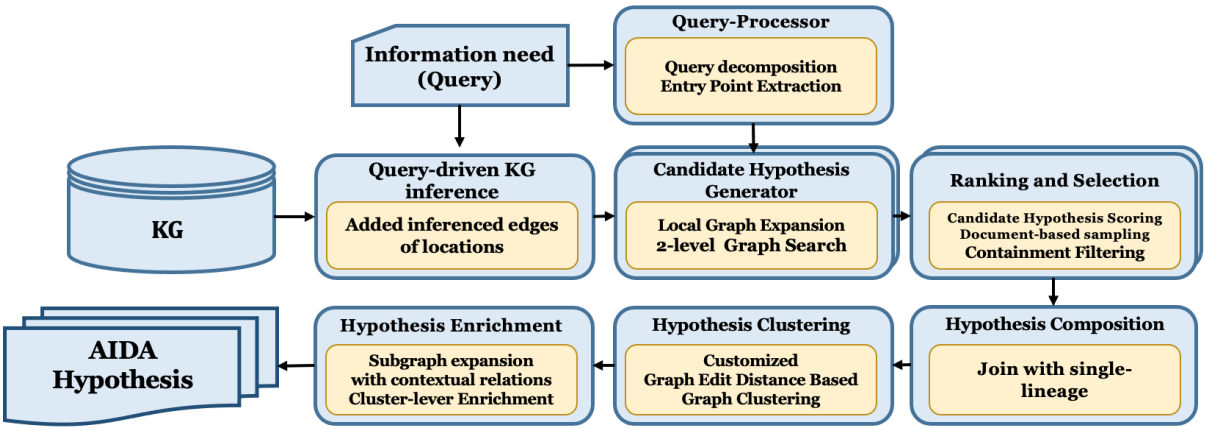

Figure 4: HypoGator System Architecture

an adaptive threshold strategy are used. By dropping the duplicated entity mention nodes, we get the final seed entity set of the KG.

6.2 Query-driven Knowledge Graph inference

Planned Objective: The goal is to enrich the TA2 KG in a targeted and computationally efficient manner to support coherence of a generated hypothesis.

Current Status: General inference approaches over larger knowledge graphs require heavy computational cost, this The cost is two-folds: 1) when performing inference over the whole KG, and 2) an after effect where the reset of the system needs to process the even larger KG+inferred edges. Thus we propose using the query for targeted inference on only relevant subgraphs. We experimented with both relation based and event-role based inference. In relation based inference, we limit to the relations that appear in the query (statement of information need) and filter candidate entities/fillers based on constraints in the query, e.g., entity type, entity string, etc. To enrich the TA2 KG with new relations, we employ a simple entity-partitioning and relation-scoring algorithm based on the character offsets and query constraints. To enrich event-role arguments, we filter TA1 subgraphs (documents) that include the entry points and find missing roles for every event by cross-checking the ontology. Finally we use event-type based handcrafted features (e.g., char-offset, is-entry-point, is-same-type-as-missing, string-similarity, etc.) to infer missing roles. Initial results show a significant improvement in recall over queries that we couldn't mine any hypothesis and an overall boost in recall for other queries.

with using single document lineage compared with multi-document lineage on different datasets. The result is a trade off of completeness and coherence: in M18 data single lineage generate more coherent hypotheses, in M36 preliminary results, single lineage generate very small and incomplete hypotheses. We experimented with document clustering to identify documents with similar perspectives - this generates negative results. We also worked with GAIA TA1 team to extract source of information at the document level and at the event extraction level. Currently, we are able to leverage the lineage of the source to cluster document, however, we are not able to leverage source at the extraction level due to two reasons: 1) it was not included in the M36 eval TA2; 2) it is not clear not to generate composite hypotheses using lineage at extraction level in TA3 pipeline. We will look into these challenges when we have the data ready from TA2.

6.3 Candidate Hypothesis Generation

HypoGator uses a novel two-level graph search method to generate relevant atomic hypothesis for an atomic query. Firstly, it explores the one-hop neighborhood of the seed entities at the mention level in the knowledge graph, searching for event nodes which are coherent with the given event type in the corresponding atomic query. In the meanwhile, all the argument nodes around each visited coherent event node will also be included. Every coherent event node and its argument nodes including the seed entity serve as the backbone structure for a candidate atomic hypothesis. Then

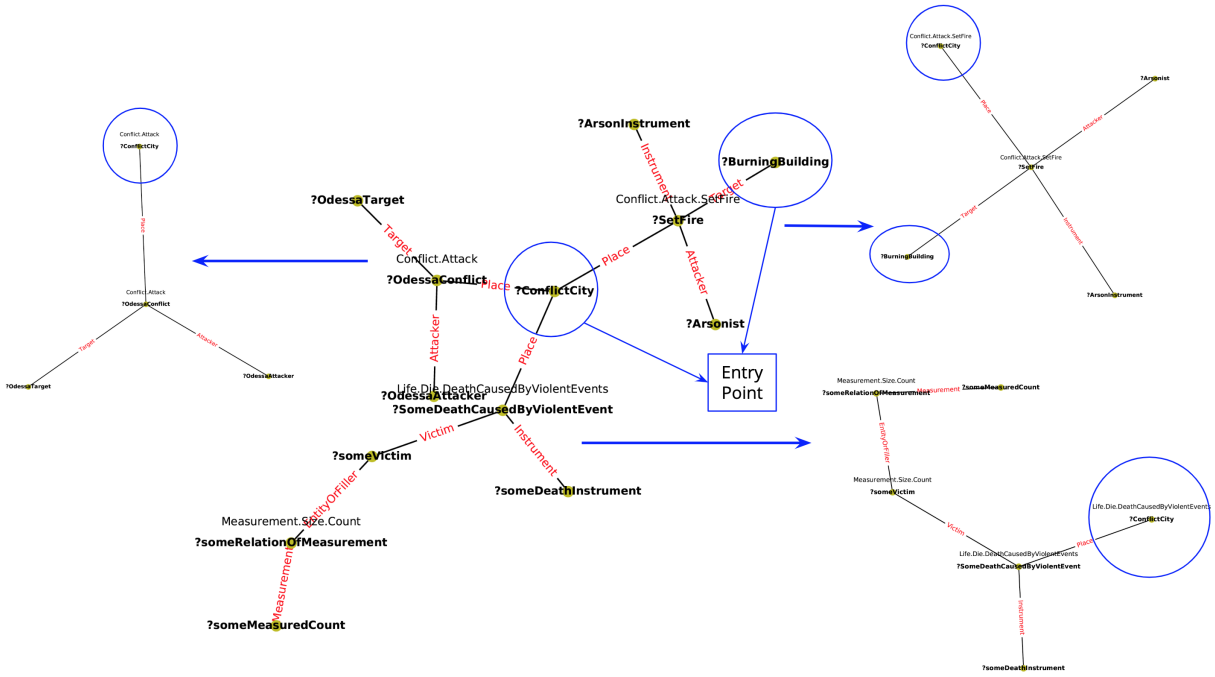

Figure 5: Query Decomposition

based on those mention level event-centric subgraphs, we continue searching for coherent relations starting from each entity around the event at the cluster level. Figure 6 gives an example of atomic hypothesis generated after the two-level graph search and context enrichment which will be introduced later.

The entity cluster information provided by TA2 increases the connectivity of the mention level graph extracted by the TA1, hence, it make us able to find more coherent information through graph searching.

6.4 Ranking and Selection

Our candidate generation module ensures that the generated candidate hypothesis include the entry points. However, this does not guarantee them to be fully relevant to the query at hand. Moreover, the candidates need to be pruned if they are not logically or semantically coherent. Another important factor determining the quality of a candidate hypothesis is the validity confidence of each of its knowledge elements, whether they are from the document sources (extraction confidence) or inferred (inference confidence) or TA2 clustering.

We use a variety of features to measure each hypothesis's semantic coherence, logical coherence, and degree of relevance to the query. We use an aggregation method to obtain an overall confidence score from each knowledge elements confidence. For example we use an ensemble of graph distance functions to measure the query relevance or use a set of predefined logical rules to detect logical inconsistency. The overall score for each hypothesis is computed as a linear combination of the individual scores from each of the features. We use the LDC labeled data to learn appropriate weights for each feature or use reasonable handcrafted weights for each feature.

While we have multiple features and each of them scores the hypothesis for some important consistency or coherence property, we need a condense score that can be used to give full quantitative significance to a hypothesis and therefore use it for ranking candidates. We use a simple approach to aggregate different scores, a weighted sum of the feature values. We manually select the weights with what we believe are more salient features of a hypothesis. We look forward to include a learned version of the weights.

6.5 Hypotheses Clustering

Due to the nature of AIDA's data e.g. multiple documents about the same hypothesis, it is possible to generate many candidate hypotheses representing the same perspective with different level of details to a given SIN. Our system uses subgraph clustering to mining out the salient alternative perspectives from the huge number of candidate hypotheses in which full of duplication and conflicts.

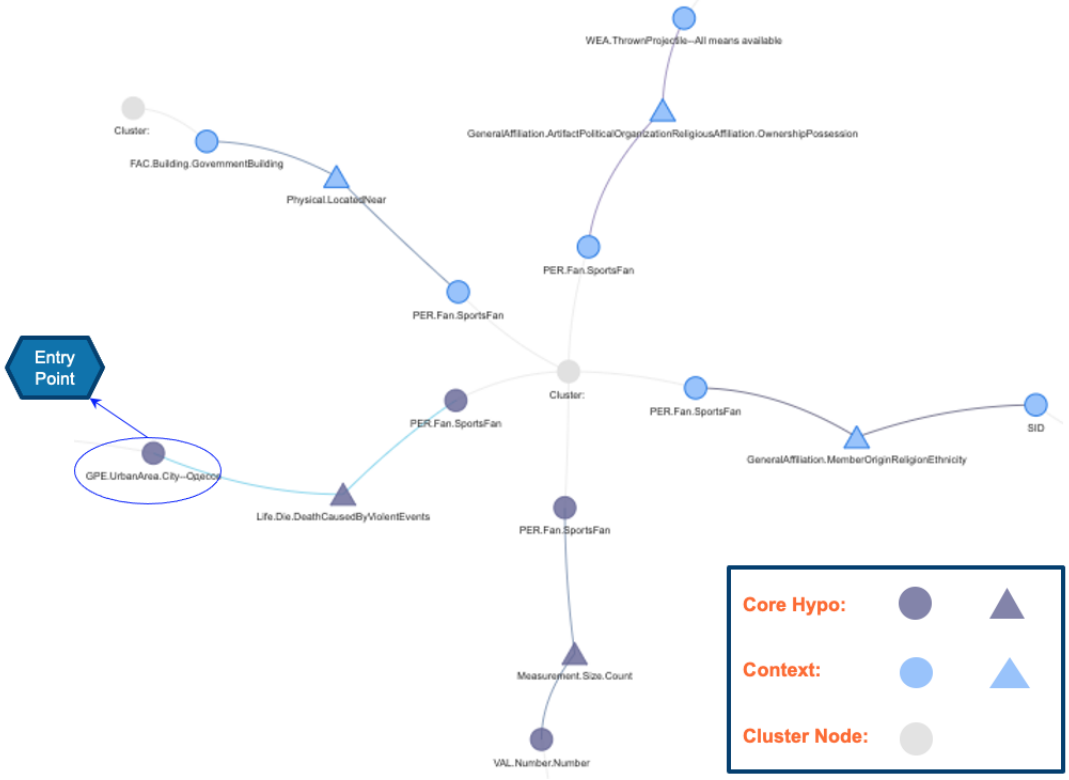

Figure 6: Example Atomic Hypothesis: the triangles in the above graph refers to event/relation nodes, the circles are entity nodes. All the purple and light blue nodes are mention level nodes, the grey circle is a entity cluster node which refers to a bunch of entity mentions(of 'PER.Fan.SportsFan' in this case) across multiple documents.

Table 1: Evaluation result of hypotheses clustering algorithms

METRICS/MOTHEDS	OLD-M18	GED-BASED(NEW)
HOMOGENEITY	0.725	0.916 (26.3(%))
COMPLETENESS	0.729	0.847 (16.2(%)
$V_measure$	0.727	0.880 (21(%))
SILHOUETTE	0.509	0.580 (14(%))
F1(REPRESENTATIVES)	0.6	0.75 (25(%))

We designed and tested five new spectral clustering based subgraph clustering algorithms with different similarity functions which is used to compute a similarity score for each pair of generated hypothesis subgraphs. To compare and evaluate these different algorithms, we manually labeled our own dataset using the subgraphs extracted from the LDC knowledge graph in which contains 54 automatically generated candidate hypothesis subgraphs and 20 manually labels clusters. Among all these new developed methods, the customized graph edit distance(GED) based one performs the best. Table 1 shows the improvement of the new GED-based method comparing to the old sting-similarity based method.

Acknowledgments

This work was supported by the U.S. DARPA AIDA Program No. FA8750-18-2-0014. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. We thank all the annotators who have contributed to the annotations of our training data for the joint IE component and keyword lists for rule-based component (in alphabetical order): Daniel Campos, Yunmo Chen, Anthony Cuff, Yi R. Fung, Xiaodan Hu, Emma Bonnette Hamel, Samual Kriman, Meha Goyal Kumar, Manling Li, Tongfei Chen, Tuan M. Lai, Ying Lin, Chandler May, Sarah Moeller, Kenton Murray, Ashley Nobi, Xiaoman Pan, Nikolaus Parulian, Adams Pollins, Kyle Rawlins, Rachel Rosset, Haoyu Wang, Qingyun Wang, Zhenhailong Wang, Aaron Steven White, Spencer Whitehead, Patrick Xia, Lucia Yao, Pengfei Yu, Qi Zeng, Haoran Zhang, Hongming Zhang, Zixuan Zhang.

References

Emily Allaway and Kathleen McKeown. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. *arXiv preprint arXiv:2010.03640*.

Amith Ananthram, Emily Allaway, and Kathleen McKeown. 2020. Event guided denoising for multilingual relation learning. *arXiv preprint arXiv:2012.02721*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Francisco Dias. 2016. Multilingual Automated Text Anonymization. Msc dissertation, Instituto Superior Técnico, Lisbon, Portugal, May.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Heng Ji, Ralph Grishman, Dayne Freitag, Matthias Blume, John Wang, Shahram Khadivi, Richard Zens, and Hermann Ney. 2009. Name extraction and translation for distillation. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Tuan Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2019. A gated self-attention memory network for answer selection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5953–5959, Hong Kong, China. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Manling Li, Ying Lin, Ananya Subburathinam, Spencer Whitehead, Xiaoman Pan, Di Lu, Qingyun Wang, Tongtao Zhang, L. Huang, Huai zhong Ji, Alireza Zareian, H. Akbari, Brian. Chen, Bo Wu, Emily Allaway, Shih-Fu Chang, K. McKeown, Y. Yao, J. Chen, Eric J Berquist, Kexuan Sun, Xujun Peng, R. Gabbard, M. Freedman, Pedro A. Szekely, T. K. Kumar, Arka Sadhu, R. Nevatia, M. Rodriguez, Yifan Wang, Yang Bai, A. Sadeghian, and D. Wang. 2019. Gaia at sm-kbp 2019-a multi-media multi-lingual knowledge extraction and hypothesis generation system.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Ying Lin, Liyuan Liu, Heng Ji, Dong Yu, and Jiawei Han. 2019. Reliability-aware dynamic feature composition for name tagging. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 165–174, Florence, Italy. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke

Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. 2016. Overview of TAC-KBP 2016 event nugget track. In *Proceedings of the 2016 Text Analysis Conference, TAC 2016, Gaithersburg, Maryland, USA, November 14-15, 2016*. NIST.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning - Proceedings of the Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes, EMNLP-CoNLL 2012, July 13, 2012, Jeju Island, Korea*, pages 1–40. ACL.

Sarah Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. 2020. Grounded situation recognition. In *European Conference on Computer Vision*, pages 314–332. Springer.

Marta Recasens, Lluís Màrquez, Emili Sapena, M. Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 1–8. The Association for Computer Linguistics.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ere: annotation of entities, relations, and events. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

Tongtao Zhang, Ananya Subburathinam, Ge Shi, Lifu Huang, Di Lu, Xiaoman Pan, Manling Li, Boliang Zhang, Qingyun Wang, Spencer Whitehead, et al. 2018. Gaia-a multi-media multi-lingual knowledge extraction and hypothesis generation system. In *Proceedings of TAC KBP 2018, the 25th International Conference on Computational Linguistics: Technical Papers.*

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.