# Panorama: BBN participation in SM-KBP 2018

**Ilana Heintz, Roger Bock, Clay Riley, Joshua Fasching, Sancar Adali, Fred Bernardin, Daniel Ellard, Gretchen Markiewicz, Kevin Jett, and John Greve**
Raytheon BBN Technologies
Cambridge, MA 02138, USA
{ilana.heintz, roger.bock, clay.riley, joshua.fasching, sancar.adali, fred.bernardin, dan.ellard, gretchen.markiewicz, kevin.jett, john.greve}@raytheon.com

## Abstract

We describe the BBN submission to the TAC 2018 Streaming Multimedia Knowledge Base Population (SM-KBP) track. We apply a variety of extraction tools to acquire information from videos, images, and text regarding persons, places, relationships, and events of interest in the targeted evaluation scenario.

## 1 Introduction

Data provided by NIST and LDC was analyzed by a series of natural language processing and image analysis modules trained on a general purpose corpora. Information in the form of "knowledge elements" relevant to the provided scenario were extracted to produce a knowledge graph intended for hypothesis generation.

## 2 Panorama Pipeline

Figure 1 depicts the Panorama pipeline, including all individual components and possible workflows. Our pilot evaluation submission was constructed by individually running each analytic in turn over the appropriate documents. Each pipeline component is run with parallel processing using automated scripts, but the initiation of each pipeline component on the appropriate data requires human intervention.

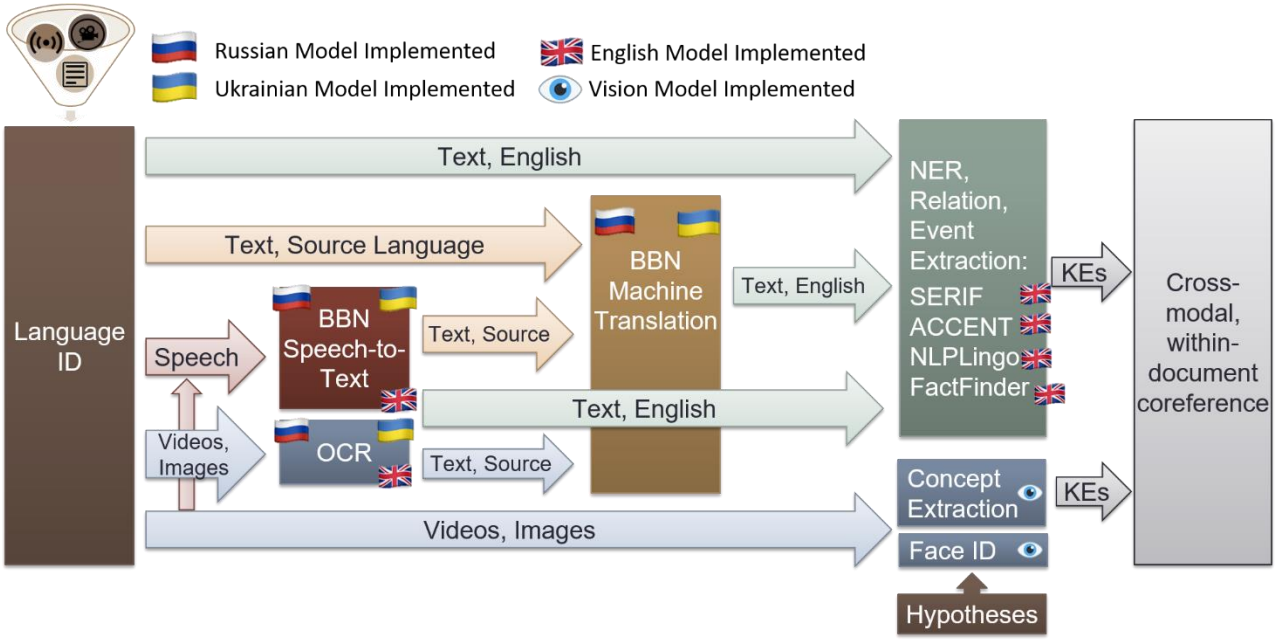


**Figure 1: Panorama Pipeline Components and Workflows**

Distribution A: Approved for Public Release, Distribution Unlimited.
This material is based upon work supported by the Defense Advanced Research Projects Agency and the Air Force Research Laboratory under Contract No. FA8750-18-C-0001. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

## 3 Language ID

For the pilot evaluation, we considered relying on the language labels assigned to each input document in the metadata provided by LDC. However, our tests on the video data found that the LDC-provided labels were unreliable, so we attempted to automatically determine the source language. For text files we used the best scoring classification produced by the open source Compact Language Detector 2 Naïve Bayes classifier[1]. When the best scoring classification was not one of the expected program languages, we fell back to using the LDC-provided language. For audio we trained an acoustic language ID model, but found it had poor performance and chose to use LDC's labels for the pilot evaluation.

To handle optical character recognition (OCR), we ran each file through our pipeline as if it were each of the three program languages. We then provided to the downstream components the extraction output from treating the document element as each of the three languages. We anecdotally found that applying text extraction to the output of OCR from the incorrect language paths produced few false positives.

In future work we plan to test whether the quantity of information extracted as knowledge elements can be used as the heuristic for choosing the correct source language.

## 4 Speech-to-Text

Automatic Speech Recognition (ASR) was used to extract a text transcript from audio files in English, Russian, and Ukrainian. In this context, the audio files were strictly the audio tracks from the provided videos. The videos were in .mp4 format, and we extracted the audio track in .wav format using the free, open source tool FFmpeg[2]. We converted the resulting .wav files to 16 KHz, 16-bit, NIST sphere format, the preferred format for BBN's speech recognizer.

For each audio file, we produce a transcript in which each word is annotated with its start time and duration. This represents our one-best output. A consensus net with all possible paths per utterance is also constructed, but was not used downstream in the pilot evaluation. The training data used to create acoustic and language models for the three evaluation languages is shown in Table 1.

We were only able to apply ASR in the unconstrained condition due to the provenance of the training data. For the evaluation, we processed 1,029 video files that contained roughly 114 hours of audio.

We tested each trained model against a target domain test set consisting of human-annotated audio files from the AIDA seedling data for each language.

| Data Type | Russian | Ukrainian | English |
|---|---|---|---|
| **Audio** | BBN broadcast news data (99 hours) | AIDA seedling audio files, human transcribed (1.4 hours) enhanced with i-vectors from Russian audio | BBN broadcast news data (1,642 hours) |
| **Text** | AIDA seedling corpus LTF documents; Russian Wikipedia | AIDA seedling corpus LTF documents; Ukrainian Wikipedia | AIDA seedling corpus LTF documents |

**Table 1: Resources for ASR training**

[1] https://github.com/CLD2Owners/cld2

[2] https://ffmpeg.org/

| Language | WER | MAP (iv) | MAP (oov) | MAP (rare) | MAP(target) |
|---|---|---|---|---|---|
| **English** | 36.10 | 0.8460 | 0.6205 | 0.8651 | - |
| **Russian** | 62.34 | 0.6154 | 0.3245 | 0.6418 | 0.5046 |
| **Ukrainian** | 65.77 | 0.5989 | 0.5194 | 0.6107 | 0.4676 |

**Table 2: Word error rate (WER) and Mean Average Precision (MAP) ASR scores on held-out seedling corpus test sets**

These data sets proved challenging due to the out-of-domain nature of the target data in comparison to the audio training data.

In Table 2, we report the overall word error rate as well as Mean Average Precision (MAP) for in-vocabulary (iv), out-of-vocabulary (oov), and infrequent (rare) terms for each language. We also measured the MAP for target words that we considered to be scenario-relevant in Russian and Ukrainian.

For English and Russian we found that there was a significant degradation when applying the acoustic models to the AIDA corpus. On a same-domain test set the English model produces closer to 11% WER, and the Russian model closer to 22% WER.

## 5 Optical Character Recognition

The evaluation corpus included text in many of the images and videos. Typically these are subtitles overlaid on the image, but this also includes ticker-style text on news programs as well as some in-scene signs. OCR is applied to transform these portions of the images into machine-readable text. This is a three-step process. First we perform text detection: the area where text appears is bounded and extracted from the image. Next, an algorithm determines whether there are one or multiple lines of text, and separates these as necessary. Lastly, the characters are matched to possible characters in the model of the specified language. The machine-readable text that is the outcome of OCR is processed by machine translation, and then by the text-based extraction components.

## 6 Machine Translation

Machine Translation (MT) was used to translate Ukrainian and Russian text into English. The resources used for training and the number of segments translated during the evaluation are shown in Table 3 .

The MT system is also capable of converting from one data format to a different format (for a limited number of formats) as required by the downstream tools. For example, text documents were provided to us in an XML document using the Logical Text Format (LTF) schema, but the next step in our processing pipeline uses plaintext, so the MT service converts from XML to text. We used the MT service to do this format conversion for the English-language inputs to streamline our intake processes, even though no language translation is required.

| Language | Resources | ASR utterances translated | OCR segments translated | Text segments translated |
|---|---|---|---|---|
| **Ukrainian** | LDC2017E06 LORELEI IL4 Incident Language Pack V2.0 Part 1 | 19K | 370K | 67K |
| **Russian** | LDC2016E95 LORELEI Russian Representative Language Pack Translation Annotation | 50K | 370K | 650K, plus 21 Russian tweets |

**Table 3: Machine translation resources and quantity of data translated**

Text (from ASR or LTF/XML) can be translated at a rate of ~35 segments per second, per language, on a 24-core machine. Translating the 650K Russian segments from the LTF input, for example, required about 5.5 hours of processing time.

## 7 Text-Based Extraction

We use a variety of supervised models to extract named entities, relations, and events from English-language text. For the pilot evaluation, all foreign language source material, including ASR and OCR output, is processed by machine translation such that we may extract knowledge elements using English-language information extraction models.

For named entity recognition, we use SERIF, which applies a discriminative Viterbi-style perceptron model to find and extract names of persons, places, and organizations [1]. Mentions are grouped into entities using a sieve-based approach [2].

SERIF also extracts a set of relations with a maximum entropy model combined with heuristics. We supplement this with a second relation finding system that applies syntactic patterns expressed using the Brandy pattern language over pairs of entities as detected by SERIF. The patterns for most of the relation classes were determined under previous projects. Patterns for remaining scenario-relevant classes were authored using examples in the seedling corpus. Using the LearnIt tool, all pairs of entities from sentences inside the English documents were identified and used as potential examples for patterns. The LearnIt tool then allows the user to query for text trigger words that identify potential relations. Scenario-relevant patterns for the relation classes were found using this tool.

We extract events using SERIF's logistic regression models as well as ACCENT, which identifies additional events and their matching arguments according to the CAMEO event ontology [3]. ACCENT finds events using structured patterns applied to augmented text graphs (normalized proposition trees that incorporate synonymy and coreference).

A third approach to extracting events, a system we call NLPLingo, also leverages SERIF's named entity extractions, but uses a pair of convolutional neural networks (CNNs) to extract trigger words and associate likely arguments with those triggers to form events. The first CNN identifies events in the text and assigns a type and a confidence to each. The second CNN ingests these events and SERIF's entities and identifies the subset of nearby entities that fill particular role slots for each event. Both networks use the word embeddings of a given chunk of the text, as well as those of the local context around candidate tokens for triggers and slot fillers. The models are currently trained on the ACE 2008 corpus, but going forward we plan to take advantage of additional annotation to incorporate new classes in the AIDA ontology.

The classes of events, relations, and entities extracted by SERIF, ACCENT, and NLPLingo are associated with the AIDA ontology via a manually-produced mapping between the ACE and CAMEO ontologies and the AIDA ontology.

## 8 Image-Based Extraction

### 8.1 Facial Recognition

Face ID was used to detect persons of interest in images (png, jpg, bmp, gif) and videos (mp4). The persons of interest were selected based on their frequency of occurrence in the seedling corpus. An open source implementation of FaceNet was adapted for this work [4].

Training occurs independently for the 3 stages of the FaceID pipeline. The three stages are: face detection, face image vector embedding,


**Figure 2: Successful facial identification of Sergey Lavrov, Viktor Yanukovych, Petro Poroshenko, and Catherine Ashton**

and face identification. The Multitask Cascaded Convolutional Network (MTCNN) face detection approach [5] was trained on the CelebA [6] and WIDER FACE [7] datasets. The learned model parameters for the MTCNN were obtained from the open source FaceNet implementation website[3]. Model parameters for the Inception Resnet v1 deep convolutional neural network architecture were trained on the VGGFace2 dataset [8].

15 different images for each of the 27 different persons of interest were obtained from the web by an in-house annotator. Face detection was run on these images and face vectors encoded. These vectors comprise the gallery. Some of the images obtained were published after 2014 when the person of interest became noteworthy.

The k-nearest neighbors within a Euclidean distance threshold are retrieved from the gallery using the FLANN library [9]. In Task 1a, majority voting based on these retrieved vectors from the gallery determined the identity of the query face vector. Figure 2 shows some examples of successful facial identification from the seedling corpus.

A confidence score was computed between the query face vector and each one of its nearest neighbors. This confidence score is the result of applying a radial basis function kernel to the query vector and a vector in the gallery. The greatest sum of these confidence scores among the different gallery identities is used to select the identity of the query vector. The average of these confidence scores, per the selected identity, is used to report the confidence in the justification. If the average confidence is below a certain threshold, the system did not add a justification for that identity.

The alternative hypothesis, for Task1b, was encoded by shortening the distance of Person entities in the hypothesis that are also in the gallery by a small constant value. This has the effect of promoting those entities being detected in cases of uncertainty. When examining the images in the evaluation corpus, 154 entities for the person of interest 'Vladimir V. Putin' were detected before applying the distance shortening. After application, 172 instances were detected. Inspection showed that most of the newly found instances were true positives.

## 8.2 Concept Detection

We trained a set of video concept detection models from open source data. Video concepts


**Figure 3: Successful concept detection of a weapon, protest, handgun, and meeting (clockwise from upper left). In the lower left photo, the man is exposing a handgun at his hip.**

[3] https://github.com/davidsandberg/facenet

may refer to contexts, objects, or situations. These are mapped to the AIDA ontology (manually) as events, relations, or entities. Some examples are shown in Figure 3.

We train multiple concept detectors using deep convolutional network models that have been fine-tuned to detect scenario-relevant concepts. For training, we use multiple datasets to form training corpora, such as OpenImages[4] and training sets curated by BBN for use in this AIDA scenario. The set of concepts are a subset of the OpenImages visual labels.

The mean average precision for the 111-concept model is 0.63. Due to large differences in label prevalence in training corpora, the confidence scores may not be accurate when this model is applied to new scenario media. Currently, the calibration leads to overconfidence of a few labels, so is not integrated in our video analytics pipeline, but we plan to adjust the calibration technique to adapt the scores for each trained model.

We first train a multi-label convolutional network (MLCN) for scenario-relevant concepts using a customized fork of the CAFFE toolbox[5], then combine MLCN detection output with the pre-trained object detection model outputs using detections for only scenario-relevant concepts. We have modified the MLCN model to be more robust to class label noise, so that mislabeled images in the training corpora do not have negative impact on model performance.

In addition to our robust MLCN model, we use a pre-trained object detection model of Atomic Visual Actions (80 actions) and a second pre-trained object detection model trained on a subset of OpenImages classes utilizing the Tensorflow toolbox. We map the concept detections to entities and other ontology elements, with bounding box and associated key-frame information. Our system currently maps detections from these model outputs to the AIDA ontology using predefined mappings. In the evaluation we processed 29,502 images and 1,029 videos with this system.

## 9 Cross-modality merging

The extractions provided by each of the text-based and image-based analytics described above are converted to the required RDF format to include name strings and appropriate justifications. At present we provide the union of all extractions for a given document (including all child documents of varying modalities) for use by TA2. Future iterations of the inter-analytic merging component of the pipeline will conduct coreference across the different document elements of a given document and combine extractions before delivery to TA2.

## 10 Query Application

### 10.1 Class Queries

A SPARQL query is constructed using the information in the query supplied by NIST. The query is applied to our knowledge graph for the document specified by NIST's query and collects the justification information associated with the class instances.

### 10.2 Zero-hop Queries

Zero-hop queries are applied as a pair of SPARQL queries. The first SPARQL query identifies all candidate nodes in the knowledge graph that satisfy the justification and type constraints of each entry point in the query. The single best match is identified among these nodes, and the second SPARQL query accumulates all of its justifications.

[4] https://storage.googleapis.com/openimages/web/index.html

[5] http://caffe.berkeleyvision.org/

### 10.3 Graph queries

Graph queries require additional processing to identify all the knowledge graph components that are requested and that are connected to the entry points. The algorithm iteratively probes the knowledge graph for edges in the query that are connected to the response subgraph in its current state, then collects justification information on its components separately.

We use the following approach to apply the given queries to the resulting knowledge graph in both the original extraction task, and the secondary task of extractions augmented by feedback hypotheses.

1) For a given query, identify all entry points. Find/resolve the 1-best match for all of the entry points to elements in our KB.

2) Look at each edge in the query that has an entry point as one of its endpoints.

    a) If the entry point was not resolved in step 1), discard the edge.

    b) Otherwise, resolve all matches for the edge's non-entry-point endpoint and for the edge itself, and associate these elements with the appropriate variable in the query.

3) For each of the remaining edges in the query which relate to only non-entry points:

    a) If the edge has previously-encountered variables as endpoints, continue.

    b) If the edge has at least one endpoint identified by a variable that was resolved above, find all matching elements for the other endpoint and the edge, associating them with the appropriate variables. Each time a query new edge is found in this way, add it as an optional clause to the SPARQL query composed to probe for subsequent edges. This binds the variable for result rows where it is newly-encountered, but leaves it blank if another row has bound it to a different, inapplicable node. The resulting query has the following components in its WHERE clause:

        i) Binding of the entry point(s) to one node

        ii) Optional clauses that contain an entry point and whose other components have been successfully matched above

        iii) Optional clauses that contain no entry point and whose other components have been successfully matched above

        iv) A clause to probe whether the current edge can be matched to any variables

    c) Repeat until no more query edges appear in step 3b.

4) Now that every variable in the query is either resolved to one or more KB nodes or not resolvable, collect the following information:

    a) Make a list of all valid mappings from pairs of query edge IDs and variable names to the nodes themselves (or null, where appropriate).

    b) Get justifications for each knowledge element: get the 1-best (subjects/objects) or 2-best (edges) justifications associated with the node on the basis of confidence.

5) Build the XML output by composing the mappings with their justifications.

## 11 Conclusion

For the SM-KBP track of TAC 2018, BBN produced a set of knowledge graphs consisting of elements drawn from text, video, and audio sources using a variety of analytic components trained on open-source and curated scenario-relevant resources. Further work includes the implementation of a more sophisticated cross-modal merging algorithm that will combine entities, relations, and events believed to refer to the same real-world entity, using information and confidence from each modality and pipeline component to support the complimentary extractions.

## 12 Bibliography


[1] L. Ramshaw, E. Boschee, M. Freedman, J. MacBride, R. Weischedel and A. Zamanian, "SERIF Language Processing - Effective Trainable Language Understanding," in *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, Springer, 2011, pp. 626-631.

[2] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu and D. Jurafsky, "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task," in *Proc. 15th Conf. on Computational Natural Langauge Learning: Shared task*, 2011.

[3] E. Boschee, P. Natarajan and R. Weischedel, "Automatic Extraction of Events from OpenSource Text for Predictive Forecasting," in *Handbok of Computational Approaches to Counterterrorism*, 2012, pp. 51-67.

[4] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *arXiv:1503.03832*, 2015.

[5] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Neural Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.

[6] Z. Liu, P. Luo, X. Wang and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proc. Int'l Conf. on Computer Vision (ICCV)*, 2015.

[7] S. Yang, P. Luo, C. C. Loy and X. Tang, "WIDER FACE: A Face Detection Benchmark," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A dataset for recognising face across pose and age," in *Int'l Conf on Automatic Face and Gesture Recognition*, 2018.

[9] M. Muja and D. Lowe, "Scalable Nearest Neighbor Algorithms for High Dimensional Data," *Pattern Analysis and Machine Intelligence*, vol. 36, 2014.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *arXiv:1408:5093*, 2014.

[11] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan and K. Murphy, "OpenImages: A public dataset for large-scale multi-label and multi-class image classification," Dataset available from https://storage.googleapis.com/openimages/web/index.html, 2017.

[12] D. Sites, "CLD2," https://github.com/CLD2Owners/cld2, 2013.