Basis Technology at TAC 2012 Entity Linking

James Clarke Yuval Merhav Ghalib Suleiman Shuai Zheng David Murgatroyd
Basis Technology Corp.
One Alewife Center
Cambridge, MA 02140, USA
{jclarke|yuval|ghalib|dmurga}@basistech.com
szheng5@emory.edu

Abstract

Basis Technology participated in the TAC Entity-Link task of the Knowledge Base Population track at TAC 2012. We developed a supervised learning approach which produces a single model that is capable of operating across all languages and entity types. Our features are based on the outputs of other models, many of which are unsupervised.

1 Introduction

Basis Technology participated in the TAC Entity-Linking task of the Knowledge Base Population track at TAC 2012. The system is an extension of our in-house cross-document coreference (CDCR) system. The main attributes of our system are:

- a single model for all languages and entity types (and the NIL cluster). Adaptive feature space for each language and entity type.
- efficient supervised training. We frame the problem as a structured prediction problem thus avoiding unbalanced data issues.
- online. We do not need to receive all documents (or queries) ahead of time.
- vertical scalability. A candidate selection phase and Apache Lucene-based implementation have enabled processing in the order of one million documents on a single node.
- unlexicalized features. Our features are based on the output of other models, some of which are unsupervised.

2 Cross-document Coreference

The cross-document coreference task can be defined as: let X denote the collection of in-document coreference chains from multiple documents. The goal is to determine whether two chains $x_i, x_j \in X$ refer to the same entity. The key challenges in CDCR are ambiguity and variety. Ambiguity arises because many entities can be referred to by the same string. Variety describes the fact that a single entity can be referred to in multiple ways.

We model CDCR as a clustering problem. We wish to identify sets of in-document coreference chains that refer to the same entity. The identities and number of entities is not known ahead of time. Typically the number of chains will be large as they are taken from a large document collection[1], thus most agglomerative clustering algorithms are inappropriate due to vast number of chains. Performing pairwise comparisons would require quadratic time and space.

Another consideration is that for CDCR to be practical it must support receiving documents in an online (or streaming) manner. It is unrealistic to expect all documents to be received in a single batch. In order scale and support the streaming scenario we use an incremental clustering algorithm in which decisions are made

[1] We observe approximately 30 in-document chains per document in Gigaword

as soon as a document arrives for processing. More formally: given an in-document coreference chain x and a set of clusters Y, the goal is to determine which cluster $y \in Y$ to place x or to create a new cluster y' with the singleton x. Initially Y is empty, but grows as the system processes new documents.

Note that we do not currently perform reclustering as upstream applications may not be able to handle chains moving between clusters.

Entity Linking For the entity-linking task we pre-seed the initial set of clusters. One cluster is created per knowledge base entry. Each pre-seeded cluster contains a single in-document chain automatically extracted from the knowledge base (see Section 4 for more details).

2.1 Learning and Inference

We cast the incremental clustering problem as a structured prediction problem, where the goal is to assign the correct cluster (or new cluster) to the in-document coreference chain. The structured prediction approach has several advantages over viewing the problem as binary classification in which the goal is to predict merge or not merge for each chain-cluster pair. The structured prediction framing does not suffer from unbalanced data as all negative examples are implicitly encoded. We also explicitly consider the new cluster decision which removes the need to tune threshold for new cluster generation.

We formulate the inference problem as:

$$\hat{y} = F_w(x, Y) = \arg\max_{y \in Y} \mathbf{w}^T \Phi(x, y) \qquad (1)$$

Where Y is the set of existing clusters (we also include a dummy new cluster symbol). Φ is a feature function which describes the relationship between the coreference chain x and the candidate cluster y. $\mathbf{w}$ is a weight vector containing the parameters of the model.

The weight vector is learned using a structural support vector machine (we use the JLIS library (Chang et al., 2010)). The correct cluster, y^*, should score higher than all other possible clusters plus the loss of scoring the incorrect cluster higher than y^*. We define our loss function as:

$$l(y, y^*) = \begin{cases} 1, & y \neq y^* \\ 0, & y = y^* \end{cases} \qquad (2)$$

Note that this can be viewed as solving a ranking problem with a slightly different loss function (similar to Dredze et al. (2010)).

3 Model

We proceed in three stages. First we build a representation of the in-document coreference chain, then we generate a set of candidate clusters and finally generate features for the chain and candidate clusters and perform inference.

3.1 Representations

We first describe the representation of an in-document coreference chain and cluster. Recall that we calculate the feature vector between a chain and an existing cluster, $\Phi(x, y)$.

We associate the following attributes with an in-document coreference chain, x:

- entity type (we only process person, organization and location entity types).
- longest mention (head mention) of x.
- adjacent entity context. The named entity immediately to the left and right of the longest mention.
- term context. The ten terms to the left and ten terms to the right of the longest mention. Not crossing sentence boundaries.
- entity context. The five entities to the left and five entities to the right of the longest mention (not including mentions from the same coreference chain).
- document term context. The terms in the document.
- document entity context. The entities in the document.

A cluster, y, is represented by the set of coreference chains it contains. We also associate the following information with a cluster:

- cluster entity type. The entity type of the coreference chain that formed the cluster.
- a cluster label. The most frequent longest mention in the cluster (ties are broken by mention length)

When a cluster is pre-seeded using the knowledge base, the following information is also captured from the structured information (Figure 1):

- The terms from a pre-defined set of facts. Example facts include: *founder* and *headquarters* for organizations; *state* and *district* for locations; and *occupation* and *title* for person entities.
- Disambiguation context. KB entries contain disambiguation context to distinguish them from other entities with the same name (e.g., *Mike Quigley (footballer)*). We apply a few simple heuristics to extract this context from the KB names.
- Anchor information from Google Crosswiki data (see Section 3.2.3).

3.2 Candidate Selection

In theory, every coreference chain that enters the system could be clustered with any of the existing clusters (or form a new cluster). However, for large data sets considering every cluster is prohibitively expensive. Rather than perform inference over the entire set of clusters we first generate a set of candidate clusters. Candidate selection has been a popular method in previous TAC systems (McNamee et al., 2011; Dredze et al., 2010). We perform candidate selection using a variety of filtering techniques that are tuned for high recall while still dramatically reducing the set of clusters to consider.

3.2.1 New Cluster Filter

The New Cluster Filter adds the dummy new cluster symbol to the candidate clusters. If this filter is not included then every coreference chain must be merged with an existing cluster. Conversely, including this as the only filter forces every coreference chain to form a new cluster.

Name 1	Name 2	Score
George Bush	George Bush	1.00
george bush	George Bush	0.99
Gorge Bush	George Bush	0.96
George W. Bush	George Bush	0.94
Bush	George Bush	0.84
George	George Bush	0.73
Jeb Bush	George Bush	0.56
Bill Clinton	George Bush	0.41
Laura Bush	George Bush	0.22
U.S.	United States	0.74
希蒙・佩雷斯	Shimon Peres	0.77

Table 1: Rosette Name Indexer similarity scores.

3.2.2 Name Similarity Cluster Filter

The Name Similarity Cluster Filter produces candidate clusters containing similar mentions to the current longest mention. We use Basis Technology's Rosette Name Indexer (RNI) which provides a similarity score between two name strings. RNI has multilingual functionality which is capable of comparing names from different languages and scripts. Similarity scores range between 0.0 and 1.0, where 1.0 indicates an exact match (see Table 1 for examples).

3.2.3 Anchor Text Cluster Filter

The Anchor Text Cluster Filter produces candidate clusters containing mentions that were seen in Wikipedia or the Web as anchor text pointing to a KB entity. This data was obtained from the Google Cross-wiki dataset (Spitkovsky and Chang, 2012).

For example, Newcastle United F.C.[2] is the largest cluster from the 2012 English-English evaluation dataset. Figure 2 shows the mentions from the cluster.

This cluster contains a lot of variation and background knowledge about the soccer team. Methods that are designed for general purpose name variation typically do not encode such entity-specific knowledge. For example, RNI does not contain the knowledge that *Toon*, *Magpies* or *Barcodes* are nicknames for the team.

[2] ⟨*id* : E0669691, *wiki_title* : Newcastle_United_F.C.⟩

wiki_title	***CERN***
type	*ORG*
id	*E0533611*
name	*CERN*
facts class	*Infobox Organization*
fact name	{*name*, European Organization for Nuclear Research Organisation Européenne pour la Recherche Nucléaire}
fact name	{*type*, Particle physics laboratory}
fact name	{*headquarters*, Geneva}
fact name	{*leader_name*, Robert Aymar}
wiki_text	*The European Organization for Nuclear Research (French: Organisation Européenne pour la Recherche Nucléaire), known as CERN (see Naming), is the world's largest particle physics laboratory, situated in the northwest suburbs of Geneva...*

Figure 1: TAC Knowledge Base entry snippet for CERN.

```
Newcastle Utd
Toon
Magpies
mags
Newcastle
Barcodes
```

Figure 2: Mentions for *Newcastle United F.C.* cluster in TAC 2012 English-English evaluation data.

```
Newcastle United
Newcastle
Newcastle United F.C.
نيوكاسل يونايتد
뉴캐슬 유나이티드
Magpies
ניוקאסל יונייטד
紐卡素
Toon Army
The Toon
http://en.wikipedia.org/wiki/
Barcode
```

Figure 3: Sample Google Cross-wiki data for *Newcastle United F.C.*

However, by leveraging the Cross-wiki data we are able to capture most of this knowledge. Figure 3 shows some of the raw information associated with *Newcastle United F.C.* in Cross-wiki data.

We filter the contents of the Cross-wiki data based on frequency and heuristics to remove common noise patterns. In addition to nicknames and alternative spellings, the anchor text also contain some cross-lingual terms.

The anchor text filter adds clusters to the candidate list if the current mention matches an anchor string.

3.2.4 Required Context Cluster Filter

The Required Context Cluster Filter removes certain candidates that passed through the other cluster filters. The idea is that some clusters are almost always accompanied by certain context, and if that context is missing, the query is too ambiguous to link to that cluster. For example, a document mentioning a city will nearly always mention the state as well. In order for a query on *Amherst* to link to the KB entry for *Amherst, New Hampshire*, we require the query context to contain the words *New Hampshire*[3]. However, there are cases where well-known cities are not always mentioned with their state, e.g. *Los Angeles*, and we make exceptions for those using probabilities from Cross-wiki data.

[3]State abbreviations are expanded to their full form during pre-processing.

3.3 Features

Our model mainly exploits unlexicalized features based on the output of other models some of which are unsupervised.

The first set of features are based on the longest mention of input coreference chain.

RNI score Two features focusing on the variety problem are derived from the RNI score. One feature captures the max (single-link) score between the input's longest mention and the cluster's longest mentions. A second feature uses the RNI score and the length of the mentions being compared. For example, when comparing *George* with *George Bush* we create a feature that indicates the comparison was between a one token name and two token name. The value of the feature is the RNI score. We bin name length into the following ranges: 1, 2, 3, 4+. This allows the learner to learn different weights for name similarity based on the length.

Google Cross-wiki Google Cross-wiki data is used for two features: (1) probability of longest mention given entity (variety), (2) probability of entity given longest mention (ambiguity).

Unsupervised ambiguity The Google Cross-wiki data is a great resource when linking to a knowledge base drawn from Wikipedia. However, in other applications it is harder to leverage. We create an unsupervised model that focuses on ambiguity. We perform in-document coreference resolution on a large corpus. From the in-document coreference chains we calculate the probability of the head mention given a mention (Equation (3)).

$$P(head|mention) = \frac{\#(head, mention)}{\#(mention)} \quad (3)$$

The feature uses the cluster label as the head mention and the longest mention of the input to be clustered as the mention.

Context features The remainder of our feature space uses an average-link normalized TF-IDF similarity comparison of the context fields associated with an in-document coreference chain and cluster. For an input x with context field q and candidate cluster y with context field r, the similarity is calculated as follows:

$$\min\left(1.0, \frac{\sum_{d \in y} \sum_{t \in x_q} \text{tf}(t, d_r)\text{idf}(t)}{|y| \sum_{t \in x_q} \text{tf}(t, x_q)\text{idf}(t)}\right)$$

TF-IDF similarity between the following context fields (see Section 3.1) are used as features:

- in-document chain entity context with cluster entity context, KB context (a combination of KB facts and disambiguation context), and cluster document term context.
- in-document chain term context with cluster term context, KB context, and cluster document entity context.
- in-document chain adjacent entity context with cluster adjacent entity context, and KB context
- in-document chain document term context with cluster document term context
- in-document chain document entity context with cluster document entity context.

3.4 Feature Binning

Previous work shows that the performance of many models can be improved using feature binning (Dougherty et al., 1995). However, determining the range of each feature bin can be difficult. We use k-means clustering to cluster all observed values for a feature using k to specify the desired number of bins. As the feature values are one dimensional we use an optimal dynamic programming implementation of k-means (Wang and Song, 2011). Features are re-normalized by the bin's upper-bound to allow the learner to discriminate between instances that fall within the same bin.

3.5 Adaptive features for all languages and entity types

Due to the different characteristics of different entity types (e.g., different number of unique

names), many of last year's systems classified the queries into three entity types first and then trained models separately for each entity type (Ji et al., 2011).

We take a different approach. Rather than applying different models to different types and languages, we duplicate our feature space based on the source document's language and the entity type agreement of the mention and cluster. This is a technique used in domain adaptation (Daumé III, 2007). For every feature we make the following versions: a general version, a version prefixed with the in-document chain's source document language, a version prefixed with the chain's type and candidate cluster type, and a version with both the language and type prefix. This allows us to learn one model that is capable of handling all languages and types.

4 Implementation Details

In this section we describe a number of implementation details relating to our pipeline and how we process the TAC data.

Documents are pre-processed using Basis Technology's Rosette Linguistic Platform (RLP). In particular Rosette Base Linguistics (RBL) is used for sentence boundary detection, tokenization and lemmatization; and Rosette Entity Extractor (REX) for detecting mentions of entities and performing in-document coreference resolution.

As previously mentioned, Rosette Name Indexer (RNI) provides name similarity scores. RNI integrates directly into Apache Lucene to provide efficient name similarity over large knowledge bases. Our cross-document coreference system also leverages Lucene in order to store state and efficiently generate candidate clusters and features.

4.1 Query Extension

TAC specifies queries as sub-strings within a document. We use a number of techniques to extract the in-document chain used for linking given only the query and document. If the query exactly matches a named entity identified by REX then we use the in-document chain and type from REX (unless the type is not person, organization or location). When an exact match cannot be found we override REX forcing the query to be annotated as a named entity. The entity type is determined using a large corpus automatically annotated by REX.

Entity Type	Frequency
Person	114,523
Location	116,498
Organization	55,813
Unknown	531,907

Table 2: Distribution of entity types in the TAC KB.

4.2 Knowledge Base Mention Identification

In order to use our cross-document coreference system for entity linking we pre-seed the system with clusters generated from the KB. A pre-seeded cluster contains a single in-document coreference chain which represents the entity. Finding a good chain to represent the entity is important as the majority of information for linking is present in the chain. A KB entry is a set of structured and unstructured information (Figure 1). We use the unstructured text snippet to find a representative in-document chain. The in-document chain to represent the cluster is selected from the unstructured text snippet. Selection is based on matching against the KB's *name*, falling back to *fullname* and other fields when matches cannot be found.

4.3 Knowledge Base Type Identification

The TAC knowledge base contains 818,741 entities. While some of these entities have been annotated with type information, many have not (see Table 2). Our feature space contains type agreement information between knowledge base and query, thus it is advantageous to have types assigned to the KB entities.

We trained a Naive Bayes (NB) classifier to predict person, location, organization, or miscellaneous types for KB entities. The classifier's features are derived entirely from the TAC KB. The classifier is only used when the entity's

Fact Class	Frequency
infobox settlement	95,142
infobox album	72,987
infobox musical artist	32,442
infobox film	27,914
infobox actor	23,077
infobox single	21,193
infobox book	15,498
infobox football biography	14,567
infobox person	12,849
infobox radio station	12,642

Table 3: Most frequent fact classes in the TAC KB.

type is not annotated in the TAC KB or in the training used by the classifier. All entities determined to be miscellaneous are removed from the KB, reducing the KB's size from 818,741 to approximately 550,00 entities.

Approach Every entry in the KB contains a class (e.g., *album*); Table 3 lists the ten most frequent[4]. We exploit the fact that the majority of classes are unambiguous (i.e., all entries belonging to a certain class are of the same entity type) and design the classifier to predict types for classes rather than entries.

We manually annotated 300 classes with one of the four entity types and automatically extracted the following features:

- **Most frequent fact fields**: Classes of the same type share many fact fields (e.g., many person classes contain a *birthdate* fact). Since facts can be noisy, we only use the most frequent ones per class; we found empirically that taking the top sixty performs well.
- **Fact class tokens**: Many classes of the same type share tokens (e.g., *Infobox chess player* and *Infobox poker player*). We include the individual tokens besides *Infobox* as features.
- **Fact class suffix**: Classes of the same type that do not share any class tokens may still have a useful suffix in common (e.g., *boxer* and *player*). We include the last two letters of each fact class as a suffix feature.

All features are binary and without smoothing. Class tokens and suffix features are prefixed using unique tokens to distinguish them from other features. Table 4 shows a few examples of features extracted for different classes – we can see that person classes share many features. The classifier achieved 95% accuracy using 10-fold cross validation on the training data. The majority of errors are miscellaneous entities being classified as person, and organization as location. We also manually fixed a few errors found in the KB, such as a few U.S. states entries associated with *U.S. state symbols* instead of *U.S. state*.

4.4 Multilingual Adjustments

Our feature space is designed to handle multiple languages, however the models that support our features can benefit from additional knowledge about the target languages. This year we focused on Chinese using an English only knowledge base.

4.4.1 Candidate Selection

RNI has built in support for name similarity between English and Chinese names. It is flexible and can be augmented with a dictionary of overrides for English-Chinese name pairs. We built a dictionary using the LDC's Chinese-English Name Entity Lists v1.0[5] which contains over 500,000 name pairs (e.g., ⟨加拿大, Canada⟩). Approximately 65% of the Chinese queries in TAC 2011 Chinese-English training and evaluation sets are covered by the corpus. Also, recall the anchor text filter (Section 3.2.3) covers entries in multiple languages.

4.4.2 Context Comparison

Our context comparison uses a vector space model (see Section 3.3). To compare context across languages, specifically across Chinese and English, we used Rosette Name Translator (RNT) to translate entity contexts from English

[4]Frequency depends on the normalization method.

[5]LDC2005T34 (Linguistic Data Consortium, 2005)

Fact Class	Features
Infobox district cambodia	SUFFIXia, INFOBOXdistrict, INFOBOXcambodia, province, name, geocode, populationasof, communes, villages, khmer
Infobox actress	SUFFIXss, INFOBOXactress, occupation, birthdate, birthplace, name, spouse, birthname, location, deathdate, deathplace
Infobox artist	SUFFIXst, INFOBOXartist, name, birthdate, field, nationality, location, deathdate, deathplace, training, movement, birthname, works
Infobox engineer	SUFFIXer, INFOBOXengineer, name, birthdate, deathdate, nationality, birthplace, significantprojects, discipline, institutions
Infobox dotcom company	SUFFIXny, INFOBOXdotcom, INFOBOXcompany, companyname, foundation, url, companytype, websitetype, currentstatus, language

Table 4: Example features used for KB type identification.

Dataset	PER	ORG	LOC	ALL
English-English	0.784	0.387	0.440	0.566
English-Chinese	0.561	0.495	0.634	0.565
English-Spanish	0.826	0.521	0.454	0.595

Table 5: TAC F1 measurements for `basistech1` 2012 evaluation data (broken down by query type).

to Chinese. We combined RNT with user specified dictionaries generated automatically from the LDC's Chinese-English Name Entity Lists.

5 Training and Evaluation

We learn the weight vector for the model using a structural support vector machine. This is a supervised learning setting which requires training examples. Our training examples consist of the filtered set of candidate clusters (the truth cluster is also added if it wasn't included in the filters). Recall that the feature vector is generated between the current in-document chain and candidate cluster. Training examples are generated under the assumption that all previous decisions were correct.

5.1 Experimental Results

For our submission we trained on the TAC 2009, 2010, 2011 training and evaluation data, and 2012 training data. Table 5 shows a summary of our results. The measurements against the 2012 evaluation data are general in the mid-to-high 50s of F1. This is a surprising result as our development scores were much higher. For development purposes we trained on all of TAC 2010 and 2011 training data and tested on 2011 evaluation data. Our results before submission were 0.83 F1 and 0.74 F1 for English-English and English-Chinese, respectively.

We also submitted systems that trained on less data (starting with the latest data) with the intuition that earlier TAC data may contain more annotation errors but omit the results from this paper.

5.2 Discussion

Our measurements on the 2012 evaluation datasets were significantly lower than we expected. In order to the determine the cause we conducted a post-mortem. The conclusion is that the 2012 evaluation datasets reflect greater ambiguity and variety than the previous TAC evaluation datasets. The model's feature space is not expressive enough to discriminate between candidate clusters in the 2012 evaluation data.

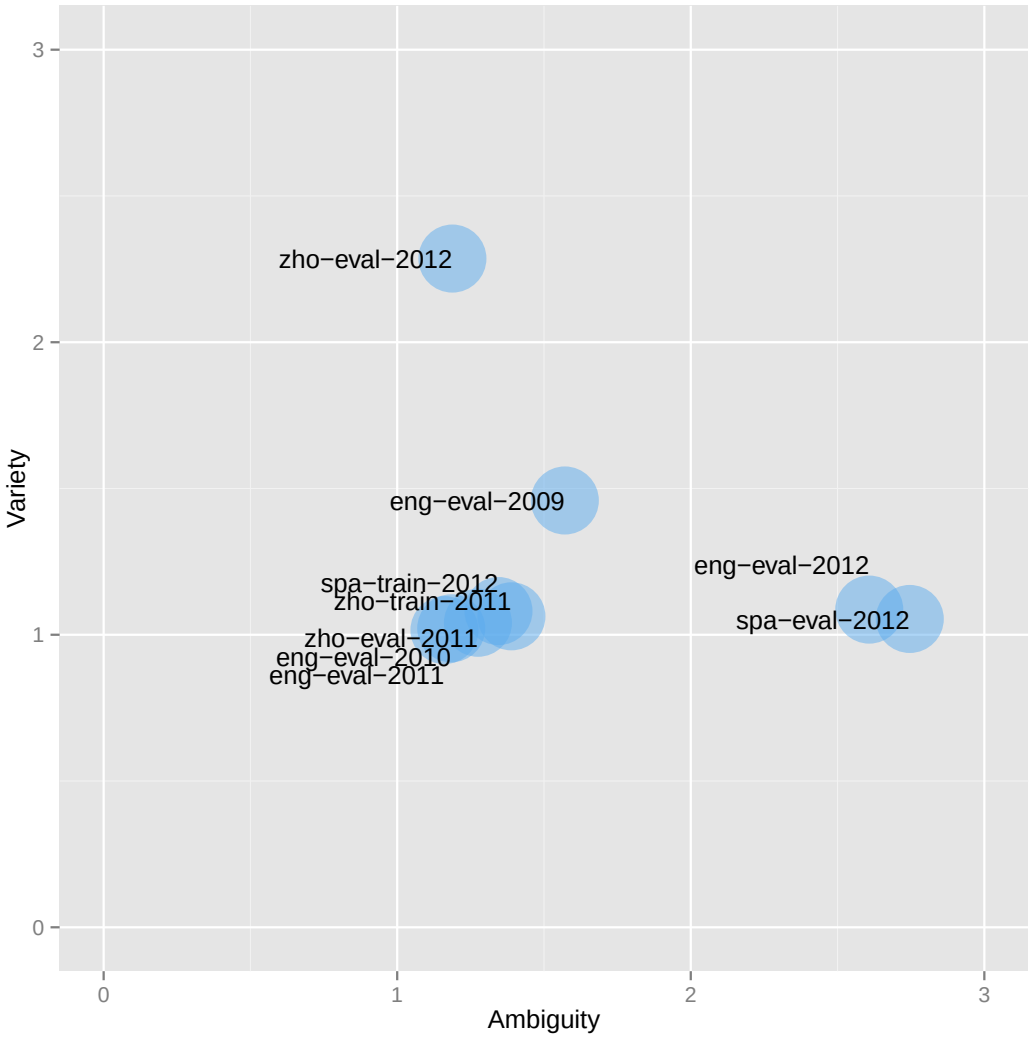

Figure 4: A comparison of ambiguity and variety across the TAC datasets.

To uncover some differences between the 2012 evaluation datasets and previous datasets, we measure ambiguity and variety of each dataset. Our definitions of ambiguity and variety differ from Ji et al. (2011). We define the ambiguity of a dataset as the average number of unique mentions per entity; and variety as the average number of entities per mention.

Figure 4 illustrates the fact that the 2012 data is both more ambiguous and more varied than previous years.

In order to better understand our model we ran an experiment with the evaluation data from each year. We measured model accuracy using 5-fold cross-validation when training on the evaluation data. Table 6 shows the results. It is clear that the 2012 data is much more difficult and our model is unable to attain the accuracy seen in previous years.

Year	Dataset	Model Accuracy
2009	English-English	86.9%
2010	English-English	85.7%
2011	English-English	86.9%
2012	English-English	69.5%

Table 6: 5-fold cross-validation model accuracy on TAC evaluation data.

6 Conclusions

Our approach to entity linking for TAC 2012 is a modification of our cross-document coreference resolution system. The system uses a machine learning approach with an adaptive feature space which allows us to learn a single model for all languages and entity types. While our system performed well on historical TAC data, it performed poorly on the TAC 2012 data. A post-mortem of the results suggests that the TAC 2012 dataset is significantly different than the data used for training. Also our feature space is not rich enough to capture the ambiguity and variety encountered in the TAC 2012 data.

Acknowledgments

This research is partially supported by the U.S. government. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of any U.S. government agency.

References

Chang, Ming-Wei, Vivek Srikumar, Dan Goldwasser, and Dan Roth. 2010. Structured output learning with indirect supervision. In *Proceedings of the International Conference on Machine Learning*.

Daumé III, Hal. 2007. Frustratingly easy domain adaptation. In *Conference of the Association for Computational Linguistics (ACL)*. Prague, Czech Republic.

Dougherty, James, Ron Kohavi, and Mehran Sahami. 1995. Supervised and unsupervised discretization of continuous features. In Armand Prieditis and Stuart J. Russell, editors, *ICML*. Morgan Kaufmann, pages 194–202.

Dredze, Mark, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *COLING*. pages 277–285.

Ji, Heng, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *TAC (Text Analysis Conference) Workshop*.

McNamee, Paul, James Mayfield, Dawn Lawrie, Doug Oard, and David Doermann. 2011. Cross language entity linking. In *International Joint Conference on Natural Language Processing*.

Spitkovsky, Valentin I. and Angel X. Chang. 2012. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*. Istanbul, Turkey.

Wang, Haizhou and Mingzhou Song. 2011. Ckmeans.1d.dp: Optimal k-means Clustering in One Dimension by Dynamic Programming. *The R Journal* 3(2):29–33.