GLOW TAC-KBP2011 Entity Linking System*

Lev Ratinov **Dan Roth**
University of Illinois at Urbana-Champaign
{ratinov2/danr}@uiuc.edu

Abstract

Traditional information extraction evaluations, such as the Message Understanding Conferences (MUC) and Automatic Content Extraction (ACE), assess the ability to extract information from individual documents in isolation. In practice, however, we may need to gather information about a person or organization that is scattered among the documents of a large collection. The TAC KBP entity linking shared task challenges the participants to identify real-word entities and to map them to a knowledgebase of reference entities. This paper describes the GLOW system which participated in the competition and was based on our earlier work (Ratinov et al., 2011).

1 Introduction

Traditional information extraction evaluations, such as the Message Understanding Conferences (MUC) and Automatic Content Extraction (ACE), assess the ability to extract information from individual documents in isolation. In practice, however, we may need to gather information about a person or organization that is scattered among the documents of a large collection. This requires the ability to identify the relevant documents and to integrate facts, possibly redundant, possibly complementary, possibly in conflict, coming from these documents. Furthermore, we may want to use the extracted information to augment an existing database. This requires the ability to link individuals mentioned in a document and information about these individuals to entries in a data base.

The Entity Linking task in KBP is formalized as follows (Ji et al., 2011). The organizers provide a set $KB = \{E_1, E_2, \ldots, E_{|KB|}\}$ of reference entities E_i, which is a subset of Wikipedia[1]. The organizers also provide a list of queries, which are tripples of the form $(Q_{id}, Q_{form}, Q_{text})$, where Q_{id} is a reference number for the query, Q_{form} is the surface form of the query, and Q_{text} is the text within which the surface form appears. Not all queries can be mapped to KB or to Wikipedia. The goal of the entity-linking task is to provide the mapping to KB for those queries which can be mapped, and (as of 2011) to cluster the rest into equivalence classes which refer to same real-world entities. Therefore, the TAC Entity Linking task is a combination of a task similar to Disambiguation to Wikipedia (Ratinov et al., 2011), (Mihalcea and Csomai, 2007), (Cucerzan, 2007), (Bunescu and Pasca, 2006), (Milne and Witten, 2008) and a task similar to cross-document co-reference resolution (Bagga and Baldwin, 1998), (Li et al., 2005). We note that we focus only on linking the queries to the TAC KBP knowledgebase. For the queries which cannot be linked, we provide a trivial solution for cross-document co-reference resolution by clustering all the queries with identical surface forms to-

* This research is supported by the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. 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 the DARPA, AFRL, or the US government.

[1] The TAC knowledge base contains 818,741 reference entities, which is about a third of 2009 Wikipedia pages.

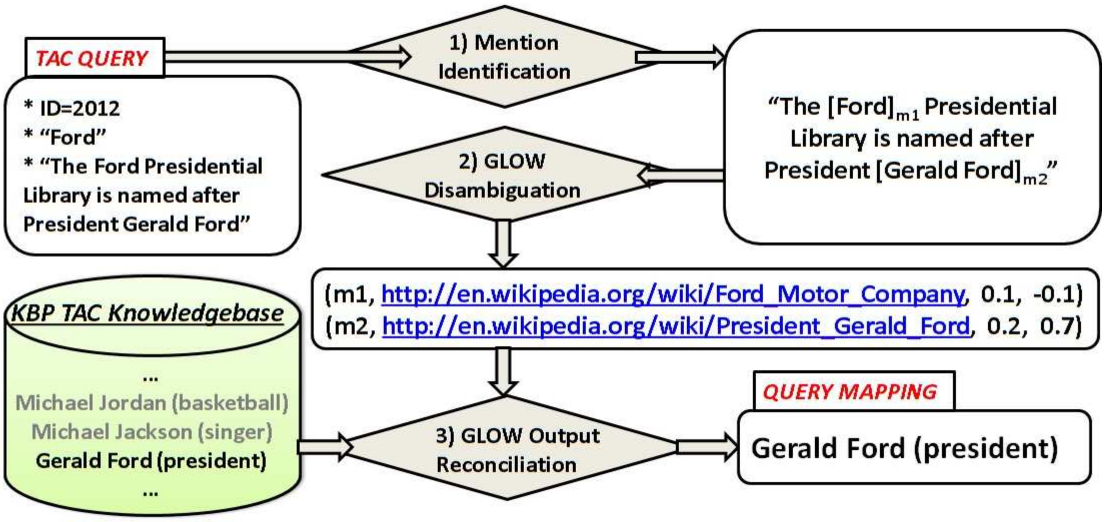

Figure 1: System architecture with an illustration of information flow. The output is provided for illustration purposes only. In reality, while the baseline model of GLOW makes a disambiguation error on $[\text{Ford}]_{m1}$, most expressive models of GLOW link $[\text{Ford}]_{m1}$ to Gerald Ford. Also, in our submitted NEQI mention identification implementation, we mark only a single mention $[\text{Gerald Ford}]_{m2}$ for GLOW as a canonical reference mention of the query. Nevertheless, the output in the figure is consistent with various flavors of our submission.

gether.

Our idea to was to use GLOW, an off-the-shelf system we have developed for a related task of Disambiguation to Wikipedia (D2W). The GLOW system takes as input a text document d and a set of mentions $M = \{m_1, m_2, \ldots, m_N\}$ in d, and cross-links them to Wikipedia, which acts as a knowledge base. This is done through combining local clues (namely lexical overlap and Wikipedia title prevalence) with global coherence of the joint cross-linking assignment (which is done by analyzing Wikipedia link structure and estimating pairwise article relatedness). The key advantage of GLOW as reported by (Ratinov et al., 2011) is using different strategies for forming an approximate solution to the input problem and using it as a semantic disambiguation context for all the mentions. This allows GLOW to maintain a tractable inference by disambiguating each mention independently while capturing important global properties. In fact, GLOW stands for Global and LOcal Wikification.

However, there are subtle differences between D2W and entity linking tasks which prevent GLOW from being applied directly. More specifically, in D2W the set of input mentions is tied to specific locations in the text, thus potentially the same surface form may refer to different entities. For example, a review about a movie Titanic may use the same surface form "Titanic" to refer both to the ship and to the movie. In D2W, each mention referring to the ship would be linked to `http://en.wikipedia.org/wiki/RMS_Titanic`, while each mention referring to the movie would be cross-linked to `http://en.wikipedia.org/wiki/Titanic_(1997_film)`. This scenario does not occur in the TAC KBP entity-linking task where one sense per document holds. On the other hand, in the entity linking task, the following query is possible (*QID, "Ford", "The Ford Presidential Library is named after President Gerald Ford"*). While in D2W, the above text would typically contain two mentions: "Ford Presidential Library" and "Gerald Ford"[2] (both of which are easy to disam-

[2] *"President Gerald Ford"* is also a possible mention; the exact mention boundaries depend on annotation guidelines. In this work we follow the CoNLL 2003 named entity recognition shared task standards and consistently exclude honorifics from

biguate), in the entity linking task it is necessary to understand that in both mentions “Ford” refers to President Gerald Ford. We note that technically, nothing precludes the D2W systems to have nested mentions such as “[The [Ford] Presidential Library]”, however most D2W systems are trained either to mimic Wikipedia link structure or to disambiguate named entities, which leads to poor performance on most nested mentions.

These differences and the choice of using a D2W component as an inference driver has dictated the structure of our entity linking system architecture, which we summarize in Figure 1. The entity-linking system is composed of the following three steps:
1) **Mention Identification**. Here we identify the mentions in the query text which correspond to the query form. We experimented with two approaches. A **Simple Query Identification (SIQI)** simply marks all the instances of the query form in the text, while a **Named Entity Query Identification (NEQI)** maps the query form to all the named entities containing the form. In Figure 1 we show the output of NEQI. The output of SIQI would be *"The [Ford] Presidential Library is named after President Gerald [Ford]"*.
2) **Disambiguation** - this step is a straighforward application of the GLOW system. We note that the GLOW system assigns each mention m_i a disambiguation Wikipedia title t_i along with two confidence scores: r_i, the ranker score and l_i the linker score. Roughly speaking, the ranker score indicates the confidence that the selected dismbiguation is more appropriate than the alternatives, while the linker score is the confidence that the mention can be mapped to the knowledgebase (in the GLOW case, Wikipedia).
3) GLOW **Output Reconciliation**. The NEQI mention matching approach has generated two mentions: *Ford* and *Gerald Ford* and GLOW has mapped them to different Wikipedia titles. We need to map the query to a single entry in the KBP knowledgebase. There are two challenges in this step: to select a single Wikipedia title as disambiguation and to map it from Wikipedia to the TAC KBP knowledgebase[3].

the named entities.

[3]The matching may not be straightforward especially if the TAC KBP was built using a different Wikipedia dump than GLOW.

The rest of the paper is organized as follows. We focus on mention identification in Section 2, on disambiguation which in Section 3, and on GLOW output reconciliation in Section 4. In Section 5 we evaluate our system on the TAC KBP 2011 shared task data and conclude.

2 Mention Identification

The goal of this step is to identify expressions in Q_{text} which are likely to refer to Q_{form}. As we mentioned eariler, we experimented with two methods: SIQI, a simple mention identification based on exact string matching, and NEQI, a named entity based mention identification with approximate string matching, which we discuss in this section.

The NEQI strategy is very similar to query expansion in many entity linking systems, for example (Chen et al., 2010). The difference is that in contrast to the traditional query expansion, the reference mentions will be bound to specific locations in the text. For example, in our running example of Figure 1, we mark the reference mentions set { *Ford*$[m_1]$, *Gerald Ford*$[m_2]$}. We note that we will disambiguate the mentions jointly, however we will ultimately allow each reference mention have a different disambiguation. Therefore, in our system even two mentions having an identical surface form could have different disambiguations. We believe that this architecture is more robust since it allows us to be more flexible in suggesting reference mentions for the query form, and to recover from potentially erroneous reference mentions. This architecture also allows us to be robust to documents which do not have a “one sense per document” property.

Below we describe the NEQI method for reference mention recommendation.

1) Annotate Q_{text} with Illinois NER tagger (Ratinov and Roth, 2009)[4]. Let $NER_{Q_{form}}(Q_{text})$ be the set of all named entities in Q_{text} which could be matched through approximate string matching to Q_{form}. Approximate string matching we applied was acronym matching (for example, *AI* would be matched with *Artificial Intelligence*) and simple rules for matching named entities, which allowed matching *Mr. Bush* to *GEORGE W. BUSH*.

[4]Available at `http://cogcomp.cs.illinois.edu/page/software`

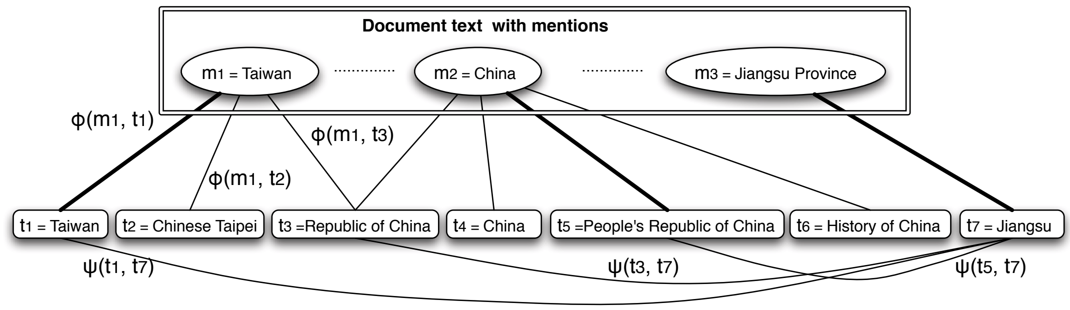

Figure 2: Sample disambiguation to Wikipedia with three mentions formalized as bipartite matching problem. The correct mapping from mentions to titles is marked by heavy edges

That is, we have simple rules for discarding professional titles, honorifics, case-insensitive matching and punctuation-insensitive matching. We note that in Figure 1 *"Ford Presidential Library"* would not be matched against *"Ford"*.

2) If $NER_{Q_{form}}(Q_{text}) \neq \emptyset$, let CF be the longest string in $NER_{Q_{form}}(Q_{text})$, let $CF = Q_{form}$ otherwise. The purpose of this step is to identify the *"canionical form"* (CF) for the query form in the text. For example, in Figure 1, { Gerald Ford$[m_2]$ is the canonical form for the query "Ford".

3) The GLOW system of (Ratinov et al., 2011) does not perform an approximate string matching, it cross-links only the expressions which appeared as hyperlinks in Wikipedia. Therefore, the GLOW would not be able to cross-link the string *LONDON* to Wikipedia. In this step, we normalize the canonical form CN to the normalized canonical form NCF. Following (Mihalcea and Csomai, 2007), we define *linkability* of an expression s as the ratio of Wikipedia pages which contain s as a hyperlink anchor to the number of Wikipedia pages which contain s in any form. For example, 1154 Wikipedia pages contain the expression *"Michael Jordan"* and out of them 959 (83%) also contain a link anchored as *"Michael Jordan"* to the Wikipedia page corresponding to the correct meaning. In constrast the expression "boarding school" appeared in 5038 Wikipedia pages, and only 1421 (28%) of them, had a hyperlink anchored with the surface from. To obtain the canonical normalized surface form for CF we compare CF against the list of all titles, redirects and hyperlink anchors in Wikipedia. We keep only those, which appeared in at least 10 Wikipedia pages and can be matched using a case-insensitive, punctuation-insensitive approximate string matching heuristis. Among the set of matched expressions, we choose the most *linkable* one.

4) If $CF \neq Q_{form}$ and CF cannot be matched to neither Wikipedia anchors nor titles nor redirects, we assume that an NER error has occurred and we revert to the normalization step with $CF = Q_{form}$.

5) We replace all the instances of CF in Q_{text} by NCF, and mark these instances as out final set of reference mentions. The GLOW system will be applied to this modified text.

3 Disambiguation

The disambiguation component of our entity linking system is performed through a straighforward application of the GLOW Wikification system of (Ratinov et al., 2011). In this section we provide a short summary of GLOW. We refer the reader to the original paper for the full details.

We formalize the task as finding a many-to-one matching on a bipartite graph, with mentions forming one partition and Wikipedia titles the other (see Figure 2). We denote the output matching as an N-tuple $\Gamma = (t_1, \ldots, t_N)$ where t_i is the output disambiguation for mention m_i. With this formulation in mind, we can write down an objective function: The common approach is to utilize the Wikipedia link graph to obtain an estimate pairwise relatedness between titles $\psi(t_i, t_j)$ and to efficiently generate a *disambiguation context* Γ', a rough approximation

Algorithm: Disambiguate to Wikipedia
Input: document d, Mentions $M = \{m_1, \ldots, m_N\}$
Output: a disambiguation $\Gamma = (t_1, \ldots, t_N)$.
1) Let M' = $M\cup$ { *Other potential mentions in* d}
2) For each mention $m'_i \in M'$*, construct a set of disambiguation candidates* $T_i = \{t^i_1, \ldots, t^i_{k_i}\}, t^i_j \neq$ null
3) **Ranker***: Find a solution* $\Gamma = (t'_1, \ldots, t'_{|M'|})$*, where* $t'_i \in T_i$ *is the best non-null disambiguation of* m'_i.
4)* **Linker***: For each* m'_i*, map* t'_i *to* null *in* Γ iff *doing so improves the objective function 1*
5) Return Γ *entries for the original mentions* M.

Figure 3: High-level pseudocode for GLOW . Step (4) is disabled for the entity-linking task.

to the optimal Γ^*. We then solve the easier problem:

$$\Gamma^* \approx \arg\max_{\Gamma} \sum_{i=1}^{N} [\phi(m_i, t_i) + \sum_{t_j \in \Gamma'} \psi(t_i, t_j)] \quad (1)$$

Where $\{m_i\}_{i=1}^N$ is the set of mentions, $\{t_i\}_{i=1}^N$ is a set of associated Wikipedia pages, ϕ is a local scoring function which assigns higher scores to titles with content similar to that of the input document, ψ is a coherence function which assigns higher scores to related titles in Wikipedia and Γ', a rough approximation to the optimal solution. We can solve the equation 1 efficiently by finding each t_i and then mapping m_i independently as in a local approach, but still enforces some degree of coherence among the disambiguations using Γ' and ψ.

The pseudocode for the original GLOW system is given in Figure 3. We note that the input document d and the mention set $M = \{m_1, \ldots, m_N\}$ which GLOW expects as input are the normalized text of Q_{text} and the set of reference mentions provided by the algorithm described in Section 2. We note that while we mark a specific set of mentions M for GLOW to link to Wikipedia, GLOW will identify and disambiguate other expressions in the input text as well, and use them as disambiguation context for disambiguating M. For our entity linking system, we disable the linker component (step 4 of GLOW). The reason is that we will form the final solution in Section 4.

4 Output Reconciliation

The given an input of a document d and a set of mentions $M = \{m_1, \ldots, m_N\}$ the GLOW system assigns each mention m_i a Wikipedia title t_i, and two confidence scores (r_i, l_i), where r_i is the *ranking confidence* that t_i is the most appropriate disambiguation among the disambiguation candidates proposed for m_i. l_i is the *linker score* indicating whether the objective function 1 would improve if we map m_i to $NULL$ instead of t_i. A positive score indicates that t_i is preferred over $NULL$ and a negative score indicates otherwise.

Given a knowledge base $KB = \{E_1, E_2, \ldots, E_{|KB|}\}$, a query $(Q_{id}, Q_{form}, Q_{text})$ and a set of tuples $\{(m_i, t_i, r_i, l_i)\}_{1\leq i\leq N}$ returned by GLOW , our goal is to assign a KB entry E^* to the query or $NULL$ if no such entry can be matched. Our first step is selecting a single Wikipedia page t^* out of the set $\{(m_i, t_i, r_i, l_i)\}$ (possibly $NULL$). We have explored four strategies:

1. **MaxNoThres** : Let
$$i^* = \underset{1\leq i\leq N}{\text{argmax}}\{r_i\} \quad (2)$$
Then $t* = t_{i^*}$. The idea is very simple: just select a title t_{i^*} which was assigned the maximum ranker confidence.

2. **MaxWithThres** : Let
$$i^* = \underset{1\leq i\leq N \wedge l_i \geq 0}{\text{argmax}}\{r_i\} \quad (3)$$
Then $t^* = t_{i^*}$. This strategy is identical to *MaxNoThres*, but we consider only the titles which were "linked" by GLOW. If GLOW assigned a negative linker score l_i to the mention m_i, we discard t_i from the list of possible results.

3. **SumNoThres** : Let
$$i^* = \underset{1\leq i\leq N}{\text{argmax}} \sum_{t_j = t_i} r_j \quad (4)$$

Mention Identification Policy	Performance			
	Micro-Average	B^3 Precision	B^3 Recall	B^3 F1
SIQI	0.752	0.709	0.740	0.724
NEQI	0.787	0.757	0.765	0.761

Table 1: The utility of mention selection. The Naive mention generation strategy is marking all the mention in query text which match exactly the query surface form. The method for mention generation proposed in Section 2 improves the micro-average performance by 3 points and the B^3 F1 by 4 points. We note that these results were obtained using the GLOW model trained on our internal newswire dataset rather than on the TAC data and with MaxNoThres solution aggregation strategy.

Then $t* = t_{i^*}$. This strategy is similar to *MaxNoThres*, except we summarize the ranker scores for all the mentions mapped to the same title.

4. **SumWithThres** : Let

$$i^* = \underset{1\leq i\leq N \wedge l_i \geq 0}{\operatorname{argmax}} \sum_{t_j = t_i \wedge l_j \geq 0} r_j \tag{5}$$

Then $t^* = t_{i^*}$. This strategy is identical to *SumNoThres*, but we consider only the titles which were "linked" by GLOW. We also summarize only over "linked" mentions.

Once we mapped the query to a Wikipedia title t^*, our next step is to map t^* to an entry E^* in the KBP TAC knowledge base. Since GLOW and KBP can use different versions of Wikipedia, we used a very recent February 2011 version of Wikipedia redirects. Therefore, a Wikipedia title t matches the KBP TAC entry E if both redirect to the same page in the February 2011 version of Wikipedia.

5 Experiments and Results

It is important to note that the ranking and the linking components of GLOW are SVM models which have to be trained. In the results reported in (Ratinov et al., 2011), we trained the GLOW system on Wikipedia articles themselves, training the system to mimic the Wikipedia annotation scheme. For the TAC 2011 entity linking task, we have also trained a GLOW model on the three publicly available newswire Wikification datasets described in (Ratinov et al., 2011) as well as a collection on 97 blogs which we Wikified using the Mechanical Turk, but have not published yet. The annotation style wad consistent to the Wikification works such as (Cucerzan, 2007) and (Milne and Witten, 2008), it was not a TAC entity linking style annotation. Overall, our "newswire" training dataset contained 200 documents. Since we never train on the TAC 2011 data, throughout this section, we directly report our results on the TAC 2011 evaluation dataset.

Utility of Mention Identification
In Table 1 we compare the performance of our submitted system with the SIQI and the NEQI mention identification policies. We note that our submitted results to TAC were obtained using a GLOW model trained on our internal newswire dataset rather than on the TAC data and with MaxNoThres solution aggregation strategy. The baseline is to mark all the mentions in the query text which exactly match the query form, we call this policy Naive mention generation. We compared the performance of the Naive strategy and the strategy discussed in Section 2, which as the table shows improves the micro-average performance by 3 points and the B^3 F1 by 4 points.

Utility of Solution Aggregation Strategies
In Section 4 we have mentioned that given a knowledge base $KB = \{E_1, E_2, \ldots, E_{|KB|}\}$, a query $(Q_{id}, Q_{form}, Q_{text})$ we use GLOW to generate a set of tuples $\{(m_i, t_i, r_i, l_i)\}_{1\leq i \leq N}$. However, our end goal is to assign a KB single entry E^* to the query, and we have suggested four approaches for generating a single solution, namely: MaxNoThres, MaxWithThres, SumNoThres, SumWithThres. In Table 2 we compare these approaches and conclude that all approaches are competitive.

Ablation Feature Study The GLOW system has several groups of features: baseline, lexical naive, lexical re-weighted, and coherence[5]. In (Ratinov et

[5] (Ratinov et al., 2011) has compared multiple approaches to capture coherence. In this work, we only report the best-performing approach: when disambiguating mention m, use

Reconciliation Policy	Performance			
	Micro-Average	B^3 Precision	B^3 Recall	B^3 F1
MaxNoThres	0.787	0.757	0.765	0.761
MaxWithThres	0.788	0.757	0.765	0.761
SumNoThres	0.794	0.763	0.773	0.768
SumWithThres	0.788	0.757	0.766	0.762

Table 2: Comparison of the solution generation policies. We note that these results were obtained using the GLOW model trained on our internal newswire dataset rather than on the TAC data. All approaches are competitive.

	Performance			
Features Used	Micro-Average	B^3 Precision	B^3 Recall	B^3 F1
Baseline	0.747	0.710	0.731	0.720
Baseline+Lexical				
Naive	0.784	0.749	0.764	0.756
Re-weighted	0.786	0.753	0.766	0.759
All Lexical	0.786	0.752	0.766	0.759
Baseline+Global				
Coherence	0.780	0.749	0.760	0.754
Baseline+Local+Global				
All features	0.783	0.754	0.759	0.756

Table 3: Ablation study of models from (Ratinov et al., 2011).

al., 2011) we ran an ablation study on the Wikification task, assessing the strengths and the weaknesses of each feature group. We concluded that the baseline features provide a very strong baseline. Lexical features lead to state-of-the-art performance, and while adding coherence features allow to further marginally improve the performance, the key difficulty was identifying when a mention refers to out-of-Wikipedia entity In other words, the linker scores are not very reliable. One of our goals of participating in the TAC KBP entity linking competition was to see whether these statements hold true for the TAC KBP entity linking task. In the following set of experiments, we have used the models obtained in (Ratinov et al., 2011) for different feature groups. All of the models were trained on around 10K paragraphs from Wikipedia articles. In Table 3 we compare the performance of the different GLOW models using different sets of features. We note that in all the experiments, we used our mention selection strategy from Section 2 and the SumNoThres single-solution generation strategy.

We make several observations. First, both the lexical features and the coherence features have improved the performance considerably over the baseline. Second, surprisingly, both the lexical and the coherence features performed extremely competitively to one another, and combining them did not lead to further improvement. Surprisingly, the naive lexical features performed almost as well as the re-weighted lexical features, which in (Ratinov et al., 2011) performed significantly better. Finally, the best configuration of models trained on the 10K paragraphs from Wikipedia articles achieved macro-average of 0.786 and B^3 F1 of 0.759, while the best configuration trained on 200 newswire documents achieved achieved macro-average of 0.794 and B^3 F1 of 0.768. Which means that a system trained on a smaller amount of newswire data and blogs marginally outperformed a system trained on a large amount of Wikipedia data. We hypothesize that the majority of test document contain enough context to easily disambiguate the mentions, as long as meaningful mentions have been identified and the correct disambiguation appears in the disambiguation candidate list.

baseline predictions for other mentions as a semantic context.

6 Conclusions

We have presented an approach for using the GLOW system for the TAC KBP entity linking challenge. Our approach was based on detecting mentions matching the query form in the query text, disambiguating them to Wikipedia using GLOW and then forming a single Wikipedia title t corresponding to the query. Finally, we have matched the assigned Wikipedia title to the KBP knowledge base using a February 2011 set of Wikipedia redirects. We noticed that although the GLOW system did not use the TAC KBP entity linking data for training or tuning, it achieved a surprisingly good performance. We noticed that matching the query form against potential mentions in the query text has a major impact on the end performance, allowing us to improve by 4 points B^3 F1 over the baseline. All reasonable strategies for reconciling potentially conflicting disambiguations for the identified mention set, such as MaxNoThres, MaxWithThres, SumNoThres, SumWithThres led to similar performance. GLOW has several feature groups. All of them performed similarly, and surprisingly a combination of multiple lexical features or lexical and coherence features together did not lead to an improvement over a single feature group. We were also surprised to discover that the GLOW system trained on 200 newswire documents outperformed the same system when trained on 10K articles from Wikipedia. Overall, the selection of the training set did not have much impact, and most of the performance gains were made through our approach for detecting mentions matching the query form in the query text and a single (either lexical or coherence) feature group. We hypothesize that the majority of test document contain enough context to easily disambiguate the mentions, as long as the correct mentions have been identified (correct being indeed matching the query form) and the correct disambiguation appears in the disambiguation candidate list.

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *ACL*.

R. C. Bunescu and M. Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*.

Z. Chen, S. Tamang, A. Lee, X. Li, W. Lin, J. Artiles, M. Snover, M. Passantino, and H. Ji. 2010. Cuny-blender tac-kbp2010 entity linking and slot filling system description. In *Text Analytics Conference*.

S. Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP-CoNLL*.

H. Ji, R. Grishman, and H. T. Dang. 2011. An overview of the tac2011 knowledge base population track. In *TAC*.

X. Li, P. Morie, and D. Roth. 2005. Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*.

R. Mihalcea and A. Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*.

D. Milne and I. H. Witten. 2008. Learning to link with wikipedia. In *CIKM*.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*.

L. Ratinov, D. Downey, M. Anderson, and D. Roth. 2011. Local and global algorithms for disambiguation to wikipedia. In *ACL*.