HITS' Cross-lingual Entity Linking System at TAC 2011: One Model for All Languages

Angela Fahrni and **Vivi Nastase** and **Michael Strube**
Heidelberg Institute for Theoretical Studies
Schloss-Wolfsbrunnenweg 35
69118 Heidelberg, Germany
{Angela.Fahrni|Vivi.Nastase|Michael.Strube}@h-its.org

Abstract

This paper presents HITS' system for cross-lingual entity linking at TAC 2011. We approach the task in three stages: (1) context disambiguation to obtain a language-independent representation, (2) entity disambiguation, (3) clustering of the queries that have not been linked in the second step. For each of these steps one single model is trained and applied to both languages, i.e. English and Chinese. A multilingual knowledge base derived from Wikipedia is at the core of the system. The results achieved in the TAC cross-lingual entity task support our one-model-for-all-languages approach: the F1 scores of all three runs we submitted exceed the median value by 4.8 to 5.5 percent points.

1 Introduction

HITS participated in the cross-lingual entity linking task at TAC 2011. Entity linking is the task of mapping text strings that refer to people, places and organizations (*query terms*) to the corresponding entry in a predefined knowledge base (*KB*). Query terms with no appropriate entry in the KB are clustered together if they refer to the same entity. Cross-lingual entity linking extends the monolingual entity linking task by allowing query terms and the respective documents to be in more than one language. At TAC 2011, the KB (henceforth *TAC KB*) is in English and derived from Wikipedia, while the queries and the respective documents are in English and Chinese.

We propose a three-step approach for cross-lingual entity linking:

1. **context disambiguation** to obtain a language-independent concept-based representation of the text documents;
2. supervised **entity disambiguation** using an SVM and a graph-based approach;
3. **clustering** of the remaining queries with no appropriate entry in the TAC KB by employing a string match approach and spectral clustering.

The main characteristic of HITS' system is the one-model-for-all-languages approach for all three steps: assuming that the knowledge sources for disambiguation and query clustering and the combination of these sources are general, i.e. identical across languages, one single model is applicable to all languages. This reduces the training costs as no new model has to be trained for additional languages and enables us to enlarge the training data set by using instances from different languages at the same time. The core of this approach builds on a multilingual knowledge base derived from Wikipedia and which is aligned with the TAC KB. The purpose of this multilingual KB is twofold: First, it enables us to overcome language specificities imposed by differences in lexicalizations in various languages. Second, it offers the possibility to extract information such as incoming or outgoing links not just from one single language version of Wikipedia, but from different ones.

The remainder of the paper is organized as follows: In Section 2 we discuss related work. Our approach is presented in Section 3, while the experiments are analyzed in Section 4.

2 Related Work

Existing approaches to disambiguate terms relative to Wikipedia do not just differ in their methods, but also regarding their aims. While some work (e.g. Csomai and Mihalcea (2008), Milne and Witten (2008)) focus on the disambiguation of a few keywords to *wikify* texts, i.e. to insert hyperlinks relative to Wikipedia, other systems (e.g. Bunescu and Paşca (2006), Cucerzan (2007) and Ji and Grishman (2011)) link named entities such as persons, places and organizations to Wikipedia. Our system attempts to solve the latter task in a multilingual context. The first module in our system, i.e. the disambiguation of the context, is related to a third group of work (e.g. Kulkarni et al. (2009), Turdakov and Lizorkin (2009), Ferragina and Scaiella (2010), Zhou (2010), Ratinov et al. (2011)) aiming to disambiguate as many strings in a text as possible relative to Wikipedia in order to obtain a semantic representation of text. A critical aspect for disambiguation is the definition of context. In contrast to features based on the surrounding words (e.g. Csomai and Mihalcea (2008) and Kulkarni et al. (2009)), concept-based features such as relatedness measures between two entries in a KB either require some already disambiguated fix points in a text (e.g. Milne and Witten (2008) or Ratinov et al. (2011)) or a global approach (Kulkarni et al., 2009). We pursue a two-pass disambiguation method similiar to the one proposed by Ratinov et al. (2011) and also experiment with a global graph-based approach.

While previous disambiguation approaches are mainly evaluated on one single language, recently released multilingual evaluation data sets[1] such as the one provided for TAC 2011 (Mayfield et al., 2011) allow to evaluate systems on several languages and to focus on portability across languages.

Recent work on entity clustering, i.e. cross-document coreference resolution, focusses on scalability. While Rao et al. (2010) propose a streaming approach that is scalable to large corpora, Singh et al. (2011) present a hierarchical model and a distributed inference technique to exploit parallelization. In contrast, we focus on cross-lingual term clustering and bypass the scalability issue by using a heuristic to narrow down the search space.

[1] http://ntcir.nii.ac.jp/CrossLink.

3 Methodology

This section describes HITS' system architecture (Section 3.1), presents our multilingual knowledge base (Section 3.2) and explains the different components in detail.

3.1 Architecture

HITS' system comprises four modules as illustrated in Figure 1. Input for the system are queries. Each query consists of a query term referring to an entity (e.g. *Hyderabad*) and a document in which the respective named entity is mentioned.

In a first step the document is **preprocessed**. The preprocessing module employs language-dependent components and performs tokenization, POS tagging and lemmatization. For English, we use the OpenNLP maximum entropy tokenizer[2] for tokenization and TreeTagger (Schmid, 1997) for POS tagging and lemmatization. Chinese word segmentation and POS tagging is performed by Stanford's tools (Tseng et al., 2005; Toutanova et al., 2003).

Next, the documents are further analyzed by the **context disambiguation module** (see Section 3.3). The aim of this step is to obtain a concept-based language-independent representation of the text documents which is used in the entity disambiguation and query clustering step. The context disambiguation module identifies terms in texts and retrieves all candidate concepts for these terms from our KB. Each term is partly disambiguated using a supervised feature-light disambiguation model so that only the most probable concepts for each term remain. As in the subsequent steps, one single model for all languages is trained.

The next module, i.e. the **entity disambiguation module**, disambiguates the query terms given the preanalyzed documents (see Section 3.4). The query terms are looked up in the KB (see Section 3.4.1). In contrast to the term identification step in the context disambiguation module more term variants are generated for the query terms to identify more candidate concepts and to improve recall. A supervised disambiguation module which uses an SVM returns for each candidate concept of a query term a confidence value (see Section 3.4.2). If the confidence

[2] http://incubator.apache.org/opennlp/index.html.

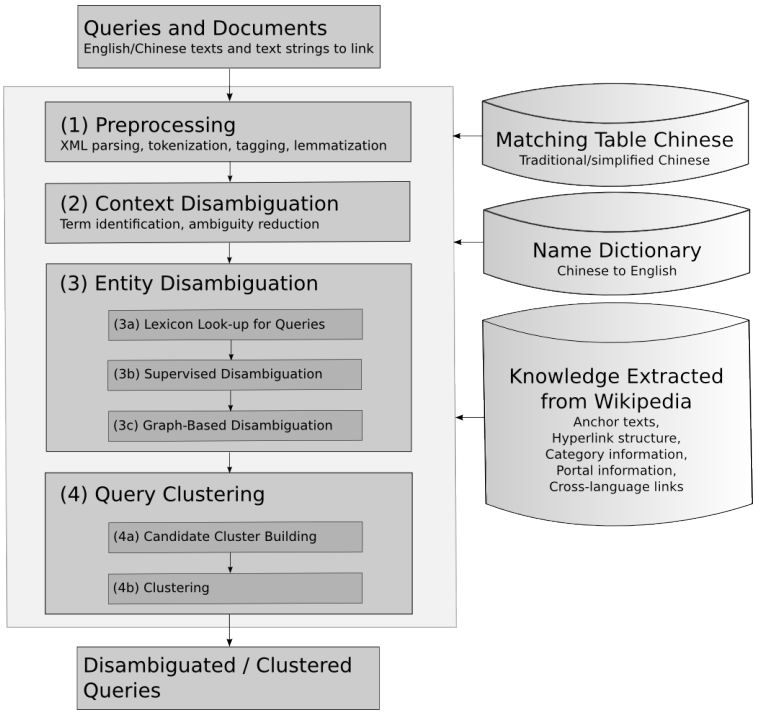

Figure 1: Architecture of the HITS system

values for all candidate concepts for a query term are below a threshold, the entity referred to by the query term is considered to be not in the KB. These queries are marked for clustering. Otherwise, if at least one candidate concept for a query term has a confidence value higher than the threshold, we assume that the entity referred to by the query term is one of these remaining candidate concepts: for the runs HITS1 and HITS2, the concept with the highest confidence score is selected, for HITS3 we applied a global graph-based approach to choose one of the remaining concepts (see Section 3.4.3).

The **query clustering module** takes as input all queries that are marked as unknown by the previous step (see Section 3.5). As a first heuristic, all queries with identical query terms or those that are translational equivalents according to a bilingual dictionary are grouped together (see Section 3.5.1). For the run HITS1, this simple string match heuristic is applied. For the runs HITS2 and HITS3, a spectral clustering algorithm is used on top of this heuristic to further partition the clusters (see Section 3.5.2).

All steps except the preprocessing step are informed by our multilingual KB. All lexicon lookup steps (2, $3a$) and the cluster step based on query term comparison ($4a$) do further benefit from a table containing mappings from traditional to simplified Chinese[3]. Additionally, a bilingual name lexicon[4] supports these steps in the runs HITS2 and HITS3.

3.2 Multilingual Knowledge Base

The core of the HITS system is a multilingual knowledge base derived from Wikipedia and aligned with the TAC KB[5]. Following previous work (e.g. Medelyan et al. (2008), Nastase et al. (2010)), we understand each Wikipedia article as corresponding to a concept while the content of the article is seen as the description of the concept. Given the English version of Wikipedia, we identify all Wikipedia articles and their corresponding lexicalizations derived from the articles' names, redirects, disambiguation pages, hyperlinks and bold terms from the first paragraph of the article. This concept repository is augmented with other information such as category associations, incoming and outgoing links to other ar-

[3] http://ishare.iask.sina.com.cn/f/6514604.html?retcode=6102.

[4] http://ishare.iask.sina.com.cn/f/15763907.html.

[5] Note, in the remainder of this paper, KB always refers to our own knowledge base.

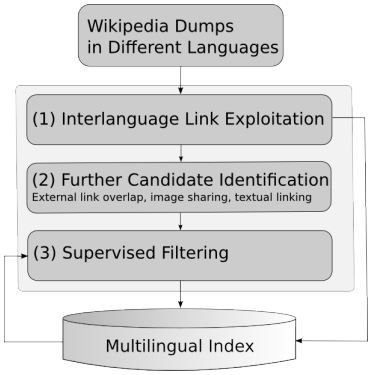

Figure 2: Building the multilingual index

ticles and word counts (see 3.2.2). To enrich this KB, we use other language versions of Wikipedia: for TAC 2011, the Chinese version is additionally processed. For each Wikipedia article in the Chinese version, we check if there exists a corresponding article in the English Wikipedia according to a multilingual index we created in advance (see Section 3.2.1). If a mapping to an English article is available, the information extracted from the Chinese version is associated with the respective, already existing concept. Otherwise, a new concept is created. To disambiguate texts in other languages than English, only lexicalizations and some statistics such as keyphraseness and prior probabilities as well as word level information (see Section 3.3) must be extracted from the respective language version of Wikipedia. Concept level information such as incoming and outgoing links can be shared across languages and can be extracted from only one or from several languages.

3.2.1 Building a Multilingual Index

A crucial step in the creation of our multilingual knowledge base is the mapping between corresponding articles in different language versions of Wikipedia. In order to obtain high coverage to avoid duplicated concepts in our KB, we proceed as depicted in Figure 2.

First, we extract all cross-language links pointing from the English Wikipedia to a target language and vice versa. The output of this step is a list of candidate mappings between English and a target language, e.g. Chinese. If there is an one-to-one mapping between a page in English and one in a target language we directly add this pair to the multilingual index. All others, i.e. one-to-many or many-to-many mappings, are appended to a list of candidate mappings. To enhance coverage, we additionally process several language versions of Wikipedia[6] and apply a triangulaton method (similar to (Wentland et al., 2008)): given three Wikipedia pages A, B and C in three different languages and two cross-language links, one pointing from A to B and one from B to C, we also establish a link between A and C. As for the direct cross-lingual links we add one-to-one mappings to the multilingual index with confidence value 1.0, one-to-many and many-to-many ones to the candidate list. To retrieve more candidate pairs, we also process other information sources which may indicate a mapping between two pages even if there exists no cross-language link:

1. **External hyperlinks**: If pages from different language versions share some external links, they are likely to deal with the same thing.
2. **Images**: Pages containing the same image tend to be similiar as well.
3. **Templates**: Sometimes the English name or word is mentioned in the Chinese Wikipedia article and the other way around using a template such as the *lang template*[7]. If we can uniquely map the name or word in the foreign language to a Wikipedia page, we also count the respective pair of Wikipedia pages as candidate pair.

In order to reduce the noise in the list of candidate mappings, a supervised filtering technique is applied: for each target language a binary classifier is trained on instances derived from the multilingual index and by using features such as a relatedness measure based on link overlap. For each English article the highest ranked mappings according to the confidence value returned by the classifier are added to the multilingual index.

Table 1 shows a quantitative evaluation of this mapping process: the first two rows indicate the coverage by using direct interlanguage links pointing

[6]We process the following language versions: English (2011/01/15), Chinese (2011/06/23), Japanese (2010/11/02), Korean (2011/06/21), German (2011/01/11), Italian (2011/01/30), French (2011/02/01), Russian (2011/07/16), Dutch (2011/01/26).

[7]`http://en.wikipedia.org/wiki/Template:Lang`.

	EN / ZH
Fract EN CLD	0.05
Fract ZH CLD	0.52
Fract EN Total	0.08
Fract ZH Total	0.59

Table 1: Coverage of the multilingual index

from the target language to the English Wikipedia and vice versa: *Fract EN CLD* and *Fract ZH CLD* exhibit the fraction of articles in the respective language with a mapping to the other language version. The last two rows (*Fract EN Total, Fract ZH Total*) report the coverage after applying the described mapping procedure.

3.2.2 Knowledge Extracted from Wikipedia

Both the disambiguation steps and the query clustering procedure are informed by knowledge derived from Wikipedia. We extract the following information from the English and partly from the Chinese Wikipedia:

Incoming links from list articles: For each concept all list articles in which it appears are identified in the English and Chinese Wikipedia.

Incoming and outgoing links: Links are extracted from the English and Chinese Wikipedia (here we do not include links from list articles as these are conceptually different, see previous item).

Categorial information: For each concept we extract all categories which do not have an administrative purpose from the English Wikipedia and also include categories that are at most three steps above in the category hierarchy.

Portal information: In Wikipedia, articles on the same subject are often summarized under a portal. There is, e.g., a portal on *martial arts*. Portals point to all relevant categories for the given subject. We extract these categories and expand them to retrieve the subsumed articles. Only the English Wikipedia is considered.

Nouns, adjectives and verbs from each article: For each article in the English and Chinese Wikipedia, all nouns, adjectives and verbs are extracted and stored together with their respective *idf* score in an index.

Nouns, adjectives and verbs appearing in the local context of internal hyperlinks: For each internal hyperlink in the English and Chinese Wikipedia all nouns, adjectives and verbs that appear in its local context are extracted and stored together with the linked Wikipedia article and the respective *idf* values in an index. The local context is defined by a context window which includes five tokens before and after a hyperlink.

3.3 Context Disambiguation

The first step after preprocessing is the disambiguation of context (see Figure 1). The aim is to obtain a concept-based language-independent representation of the documents associated with queries. This representation is used in the entity disambiguation and query clustering step. In contrast to the entity disambiguation step not only named entities, but also common nouns are disambiguated.

Given a document, all n-grams that are in our lexicon are retrieved. As anchor texts in Wikipedia are quite noisy, we only consider an n-gram as a candidate term if its keyphraseness – a metric that captures the probability of a term being linked in a Wikipedia page – exceeds a threshold[8]. For each candidate term, all candidate concepts with a prior probability higher than a certain threshold[9] are retrieved from our lexicon. The prior probability for a concept c given a term t is defined as

$$p(c|t) = \frac{count(t_c)}{\sum_{t_i \in C_t} count(t_i)} \quad (1)$$

where $count(t_c)$ is the number of times term t is linked to concept c in Wikipedia and C_t the set of candidate concepts for term t.

To identify the most probable concepts for each term, a supervised approach is pursued. A classifier is trained on instances extracted from 300 featured English Wikipedia articles we randomly selected.[10] The positive instances are derived from the hyperlinks extracted from these articles, while the negative examples are deduced by randomly selecting other candidate concepts from our lexicon for the anchors of these links. In accordance with

[8] We empirically set this threshold to $t = 0.01$

[9] We empirically set this threshold to $p = 0.01$. This filter reduces noise that results from the inclusion of anchor texts from Wikipedia in our lexicon.

[10] Wikipedia articles tagged as *featured* are supposed to be of high quality.

the one sense per discourse assumption (Gale et al., 1992), all occurrences of the candidate concept, i.e. all terms in the text that potentially refer to this concept, are considered to generate features for an instance. The features describe the following aspects:

1. The **prior probability** (Equation 1) expresses the probability of a concept given a certain term. The term *staff* for example refers more often to *Employee* ($p = 0.22$) than to the concept *Gun (staff)* ($p = 0.02$). As all occurrences of the questioned concept are considered, the average, maximum and minimum prior probability are included as features.
2. Various **string match** features capture the similarity between the Wikipedia article name of a candidate concept and the terms in a text. If article name and terms are similar, it is more likely that the terms refer to this article.
   a. Levenshtein distance is calculated using LingPipe's implementation[11]. As the scores are highly influenced by the length of the string, we normalize the scores by the number of edits necessary if the shorter string is empty. The normalized average, maximum and minimum Levenshtein distances calculated over all occurrences serve as features.
   b. If one of the terms is a substring of the respective Wikipedia article name, the value of the substring feature is 1, otherwise 0.
   c. To check if a term is an acronym of a Wikipedia article name or vice versa, some heuristics such as taking the first character of each token in a term are used. If an acronym relation is identified, the value of the acronym feature is 1, otherwise 0. For Chinese terms, the value of this feature is always 0.
3. **Context fit**: In a text about Chinese martial arts e.g., it is more probable that the anchor *staff* refers to *Gun (staff)* instead of *Employee*. The context fit is approximated on two levels:
   a. **Context fit on the conceptual level**: In order to calculate context fit on the conceptual level, some fix points are needed. While Milne and Witten (2008) use the concepts of all unambiguous terms as fix points which is problematic as it is not guaranteed that unambiguous anchors are present, Ratinov et al. (2011) employ the disambiguation results of a first disambiguation pass. We approach this problem slightly differently: for each term in a document all candidate concepts are ranked according to prior probability. The top ranked concepts that form together half of the probability mass are considered as fix points ($C_{t\_p0.5}$). Each fix point concept c obtains a weight w_c defined as

$$w_c = \frac{p(c|t)}{\sum_{c_i \in C_{t\_p0.5}} p(c_i|t)} \qquad (2)$$

where $p(c|t)$ is the prior probability of concept c given term t as defined in Equation 1.

Given these fix point concepts we build four context vectors:

i. *Category context vector:* it contains the weights for each category associated with the fix point concepts according to our category index. The weight for each category w_{cat} is determined by

$$w_{cat} = \frac{\sum_{c_i \in C_{cat}} w_{c\_i} f_{cat}}{\sum_{cat_i \in Cat} w_{cat_i}} \qquad (3)$$

where C_{cat} is the set of all fix point concepts that are associated with category cat, w_{c_i} the weight of fix point concept c_i (see Equation 2), f_{cat} the inverse frequency of category cat and Cat the set of all categories associated with at least one of the fix point concepts. For the other context vectors the weights are calculated analogously.

ii. *Concept context vector:* it consists of the weights for all fix point concepts as well as their incoming and outgoing links

iii. *List context vector:* it holds the weights of the fix point concepts' incoming links from lists.

iv. *Portal context vector:* it comprises the weights for the portals associated with the fix point concepts.

[11] http://alias-i.com/lingpipe.

For each candidate concept for a certain term, a category, a concept, a list and a portal vector is built. The weights are 1 or 0 depending on the presence or absence of the respective category, concept, list or portal in the corresponding context vector. As features we use the dot products of each vector pair and the largest and smallest summand of the respective dot products[12].

b. **Context fit on token level**: Similar to previous work (Ratinov et al., 2011; Kulkarni et al., 2009), the context fit on token level is calculated based on the whole document and on the local context using a window of five tokens before and after each occurrence. While the whole document is compared to the corresponding Wikipedia article of a candidate concept, local contexts are checked against the surrounding tokens of hyperlinks in Wikipedia that point to the candidate concept. In both cases, we calculate the cosine similarity based on nouns, verbs and adjectives separately and on all three types together. The tokens are weighted in the way described in Ratinov et al. (2011).

While features based on concepts are language-independent, features such as the ones measuring context fit on token level and string similarities use language-specific information. However, we assume that the feature values, i.e. the resulting numbers, are language-independent: the model trained on one language, namely English, is used for other languages such as Chinese.

As classifier, SVM is applied[13]. All candidate concepts for a term with an confidence score higher than a threshold, are kept[14].

3.4 Entity Disambiguation

The entity disambiguation decides whether the entity referred to by the query term is part of the TAC KB, and if so to which entry it should be linked.

[12]The contribution of the present candidate concept and competitive candidate concepts to the weights of the context vectors are ignored.

[13]LibSVM integrated in Weka is used (EL-Manzalawy and Honavar, 2005).

[14]The threshold is set to $t = 0.8$ using instances extracted from 100 featured English Wikipedia articles.

	Training Set		Test Set	
	Corr	**Amb**	**Corr**	**Amb**
EN	0.928	16.92	0.894	18.37
ZH	0.867	5.56	0.812	3.55
ZH_Lex	0.930	22.86	0.883	18.81

Table 2: Statistics after lexicon lookup on training and testing data

3.4.1 Lexicon Lookup

The purpose of the lexicon lookup is to allow for high recall without introducing too much noise in the disambiguation process.

First, n-grams ending with the query term – or in case of Chinese the simplified or traditional equivalent respectively – are extracted from the document. Then different variants of the query term are generated: lowercase and uppercase versions, versions without periods, dashes and spaces, simplified and traditional Chinese versions respectively and in the runs with bilingual lexicon an English version for Chinese terms. For all generated term variants and the ones extracted from the text document, the candidate concepts with a corresponding entry in the TAC KB are retrieved from the lexicon. If no such candidate concept is identified in case of the multi-word terms, term variants consisting of parts of multi-word terms are checked. Chinese terms are not only looked up in the Chinese, but also in the English lexicon. All queries for which no candidate concept can be found using these strategies, are marked for clustering.

Table 2 shows some statistics after the lexicon lookup step for the training and testing data separated by languages: the fraction of queries for which the correct entry in the KB is among the identified candidate concepts ($Corr$) and the average ambiguity (Amb). Coverage and average ambiguity are much higher for English (EN) compared to Chinese (ZH) which can be traced back to the size of the respective Wikipedia versions. The inclusion of the bilingual lexicon increases coverage and average ambiguity for Chinese (ZH_Lex) to a level comparable to the one for English. This module could be further improved by using more advanced techniques to generate term variations and considering additional sources such as query logs (Riezler and Liu, 2010).

3.4.2 Supervised Entity Disambiguation

For queries with at least one identified candidate concept it has to be decided whether one of the candidate concepts is the referred one, and if so which candidate concept should be selected. This task is approached in a supervised way. The training examples provided by the organizers are used to create an unbalanced set of training instances: the correct candidate concepts build the positive instances, the wrong candidate concepts retrieved in the lexicon lookup step the negative ones. The training instances derived from Chinese and English queries are not separated but build one single training set. To generate the features, all occurrences of a candidate concept in the text are considered. Besides the features already used for context disambiguation (see Section 3.3) additional, more resource-intensive features are computed:

1. **Prior probability** (see Section 3.3)
2. **String match features** (see Section 3.3)
3. **Context fit** (see Section 3.3): For the concept-based context fit features, the candidate concepts identified by the context processing component are used as fix points instead the top n ranked candidated concepts according to their prior probability.
4. **Context fit based on pairwise calculations**: These features are computationally expensive as they are calculated pairwise for each candidate concept / fix point concept pair. Hence, we use them only for the entity disambiguation step. For each pair, we calculate:

   a. **A relatedness measure based on *incoming links*** (Milne and Witten, 2008). Incoming links for a concept c_A are hyperlinks that "point to" the page corresponding to c_A. This measure captures first-order co-occurrence information at the concept-level – the more pages link to both c_A and c_B, the higher the value:

   $$rel_{in}(c_A, c_B) = \frac{log(max(|A|, |B|)) - log(A \bigcap B)}{log(|W|) - log(min(|A|, |B|))}$$

   A and B are the sets of c_A's and c_B's incoming links respectively, and W is the set of Wikipedia concepts.

   b. **A relatedness measure based on *outgoing links*** (Milne and Witten, 2008). Outgoing links for a concept c_A are hyperlinks that originate on the page corresponding to c_A. This measure captures a simplified version of second order co-occurrence information – it relies on the extent to which concepts that appear in c_A's page also occur in c_B's page:

   $$rel_{out}(c_A, c_B) = \cos(OutW_A \cdot OutW_B)$$

   $OutW_A$ and $OutW_B$ are weighted vectors of outgoing links for c_A and c_B respectively. A weight is the logarithm of the inverse frequency of the respective outgoing link: the more often a concept is linked in Wikipedia, the less discriminative it is and the smaller its weight.

   c. **A relatedness measure based on *categorial information***: Categories are assigned by Wikipedia contributors and group pages that have something in common. Hence, pages under the same category are related. We compute this relatedness measure as the cosine similarity between the vectors of the extended parent categories of concepts c_A and c_B:

   $$rel_{cat}(c_A, c_B) = \cos(CW_A \cdot CW_B)$$

   where CW_A and CW_B are two vectors containing the weights of c_A's and c_B's extended parent categories, respectively. A weight is the logarithm of the inverse frequency of the respective category. The assumption is that the less frequent a parent category is, the more informative it is if both concepts c_A and c_B are associated with it.

   d. **The preference of a concept for a context term's disambiguation.** For two terms to be disambiguated, t_A and t_B, we compute how much the disambiguation c_A for term t_A prefers the disambiguation c_B for anchor t_B:

   $$pref_{AB}(c_A, c_B|t_B) = \frac{count(c_A, c_B)}{\sum_{c_j \in C_{t_B}} count(c_A, c_j)}$$

C_{t_B} is the set of concepts that term t_B may refer to. $count(c_A, c_j)$ is the number of times the concept pair (c_A, c_j) occurs.

e. **Co-occurrence probability of two concepts** given their corresponding terms in the text t_A and t_B:

$$coocP(c_A, c_B) = e^{p(c_A,c_B|t_A,t_B)-chance(t_A,t_B)-1}$$

$$p(c_A, c_B|t_A, t_B) = \frac{count(c_A, c_B)}{\sum_{c_i \in C_{t_A}, c_j \in C_{t_B}} count(c_i, c_j)}$$

$$chance(t_A, t_B) = \frac{1}{|C_{t_A}| \times |C_{t_B}|}$$

C_{t_A} and C_{t_B} have the same meaning as above. This measure takes into account the ambiguity of the terms to be disambiguated, and quantifies the strength of the association between one specific interpretation of the two concepts considering all other options. $p(c_A, c_B|t_A, t_B)$ quantifies the absolute strength of the c_A, c_B pair, and we deduct from this the $chance(t_A, t_B)$. The reason for this is that if all concept pairs are equally likely, it means that none are really informative, and as such should have low strength. -1 is deducted to map the function to the [0,1] interval.

For each of these pairwise relatedness measures, we use the average, maximum and minimum score.

We apply an SVM in the implementation of EL-Manzalawy and Honavar (2005). To decide if a query term refers to an entity which exists in the KB or not, a threshold is set based on the training data by using five-fold crossvalidation.[15] If no candidate concept for a query term exeeds the threshold, the query is marked for clustering. Otherwise one of the candidate concepts with a confidence value higher than the threshold has to be selected. For run HITS1 and HITS2, the candidate concept with the highest confidence value is chosen, for run HITS3 a global graph-based approach outlined in the next section is used.

[15]The threshold is set to $t = 0.0859$ if a lexicon is used, otherwise to $t = 0.0719$.

3.4.3 Graph-based Entity Disambiguation

To select among the candidate concepts with a confidence value higher than a certain threshold (see Section 3.4.2), we also experimented with a global graph-based approach. Each text document is represented as a complete n-partite graph $G = (V_1, ..V_n, E)$. Each partition V_i corresponds to an term t_i in the text including the query term, and contains as vertices all remaining candidate concepts c_{i_j} for term t_i (see Section 3.3, 3.4.2). Each vertex from a partition is connected to all vertices from the other partitions through edges $e_{v_i,v_j} \in E$ whose weights w_{v_i,v_j} are determined by applying a supervised approach using the pairwise features described in Section 3.4.2 to approximate the co-occurrence probability of two concepts. In this graph we want to determine the maximum edge weighted clique.

A clique is a subgraph in which each vertex is connected to all other vertices (Newman, 2010). A maximum clique of a graph is the clique with the highest cardinality. Given our n-partite graph G a maximum clique contains for each partition (term t_i) exactly one vertex (concept). A maximum edge weighted clique is the clique C with the highest edge weights sum $W_e(C)$ (Pullan, 2008):

$$W_e(C) = \sum_{v_i,v_j \in C} w_{v_i,v_j}$$

Identifying the maximum weighted clique of a graph is an NP-complete problem, but several approximations have been proposed (see Pullan (2008), Bomze et al. (1999) for an overview). We apply an adapted beam search algorithm to approximate the maximum edge weighted clique. For each term, including the query term, the concept which corresponds to the vertex which is part of the clique in this partition is selected.

3.5 Query Clustering

The aim of this step is to cluster query terms with no corresponding entry in the KB that refer to the same entity. The way we approach this task requires a pairwise comparison of all queries to process. To keep the cost for comparison low, we use a heuristic to preselect the queries that should be checked against each other.

Section 3.5.1 describes how the preclustering is

performed, Section 3.5.2 presents the clustering approach which is applied on top of the heuristic.

3.5.1 Preclustering Based on a String Match Heuristic

To precluster the queries, a string match heuristic is used. All query terms which match each other are marked for further comparison, while minor variations such simplified vs. traditional Chinese characters or differences in capitalizations are allowed. In the runs using a bilingual dictionary (HITS2, HITS3) also translational equivalents are considered.

3.5.2 Spectral Clustering

To further partition the preclusters into subclusters and to sort out singeltons, first, a fully connected graph for each precluster is built consisting of one vertex for each query that is part of the respective precluster. Edges are weighed by a confidence score returned by a supervised model incorporating various similarity measures. For training, instances from the training data provided by the task organizers are generated: positive instances are formed by query pairs belonging to the same cluster, negative ones by pairs from different clusters. The similarity measures used as features are derived from the language-independent concept-based representations of the text documents produced by the context disambiguation component (see Section 3.3). For each query to compare, several vectors are built using the same approach to calculate the weights as described in Section 3.3. For each of the following information sources (see also Section 3.3), two vectors are created, one for the whole document, one for the local contexts of the text occurrences of the query term:[16]

1. Identified concepts
2. Identified concepts extended by incoming and outgoing links
3. Categories associated with the concepts
4. Lists in which the concepts occur in Wikipedia
5. Portals associated with the identified concepts

For each query pair to compare, the cosine similarities between these vectors are calculated and used as a feature. As a classifier SVM (EL-Manzalawy and Honavar, 2005) is applied. As all these similarity measures do not depend on a certain language, it does not matter if the two queries to compare are in the same language or not: one single model is applied to English, Chinese and English/Chinese query pairs.

Given this graph, a recursive two-way spectral clustering algorithm (Shi and Malik, 1997) which has been successfully applied to coreference resolution (Cai and Strube, 2010) is used to partition it. The parameters, i.e. the stopping criterion α^* and the parameter β controlling the singleton split, are tuned on the training data.[17]

[16]The local context window includes five tokens in front and after an occurrence of a query term.

	Micro-Average
HITS2 EN	0.788
HITS2 ZH	0.784

Table 5: Micro average scores for run HITS2

4 Experiments

We submitted three runs for the cross-lingual entity linking task at TAC 2011. Table 3 describes the differences between the runs. Table 4 summarizes the results of the three runs in comparison to the results of the best system and the median. As the numbers in Table 4 indicate, the F1 scores of all runs exeed the median by between 4.8 and 5.5 percent points. Table 5 shows that the micro average scores are similar across languages. To further evaluate our one-model-for-all-languages approach, a model for the entity disambiguation step is trained based on Chinese instances exclusively and applied to English. The micro average score for English using this model and the same setting as for run HITS2 is 0.787 and very close to the score achieved by using English and Chinese training instances (see Table 5).[18]

5 Conclusions

HITS' system for cross-lingual entity linking has proven to be successful at TAC 2011. The F1 scores

[17]α^* is set to 0.485, β to 0.01.

[18]Chinese is chosen as there are more training instances for Chinese than for English in the training set provided by the organizers.

Run ID	Approach	Resources
HITS1	Supervised entity disambiguation (Section 3.4.2) String match heuristic for entity clustering	(1) Knowledge base (see Section 3.2) (2) Stanford's Chinese segmenter and Tagger (3) Mapping table traditional to simplified Chinese (4) TreeTagger for English
HITS2	Supervised entity disambiguation (Section 3.4.2) Spectral clustering approach for entity clustering (Section 3.5)	Same as for HITS1, in addition: (5) Chinese / English lexicon
HITS3	Supervised ambiguity reduction (Section 3.4.2) Graph-based disambiguation (Section 3.4.3) Spectral clustering approach for entity clustering (Section 3.5)	Same as for HITS2

Table 3: Description of the different runs of HITS for the cross-lingual entity linking task at TAC 2011

	Micro-Average	**Precision (B^3)**	**Recall (B^3)**	**F1 (B^3)**
Best System				0.788
Median				0.675
HITS1	0.783	0.694	0.763	0.727
HITS2	**0.785**	**0.700**	**0.763**	**0.730**
HITS3	0.778	0.692	0.756	0.723

Table 4: HITS' performance compared to the best and median scores in the cross-lingual entity linking task

of all runs are well above the median value. The system implements a one-model-for-all-languages strategy with a multilingual knowledge base extracted from Wikipedia as core. The advantage of the pursued strategy is that no additional model has to be trained for any further language.

Acknowledgments. We would like to thank our colleague Jie Cai for help with regard to spectral clustering and Chinese. This work has been partially funded by the European Commission through the CoSyne project FP7-ICT-4-248531 and the Klaus Tschira Foundation.

References

Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. 1999. The maximum clique problem. In D.-Z. Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, Boston, Mass.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics,* Trento, Italy, 3–7 April 2006, pages 9–16.

Jie Cai and Michael Strube. 2010. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 143–151.

Andras Csomai and Rada Mihalcea. 2008. Linking documents to encyclopedic knowledge. *IEEE Intelligent Systems*, 23(5):34–41.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning,* Prague, Czech Republic, 28–30 June 2007, pages 708–716.

Yasser EL-Manzalawy and Vasant Honavar, 2005. *WLSVM: Integrating LibSVM into Weka Environment*. Software available at `http://www.cs.iastate.edu/~yasser/wlsvm`.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of the ACM 19th Conference on Information and Knowledge Management (CIKM 2010),* Toronto, Ont., Canada, 26–30 October 2010, pages 1625–1628.

Willian A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One sense per discourse. In *Pro-*

ceedings of the DARPA Speech and Natural Language Workshop.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics,* Portland, Oreg., USA, 19–24 June 2011, pages 1148–1158.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* Paris, France, 28 June – 1 July 2009, pages 457–466.

James Mayfield, Dawn Lawrie, Paul McNamee, and Douglas W. Oard. 2011. Building a cross-language entity linking collection in 21 languages. In *Proceedings of the 2011 Conference on Multilingual and Multimodal Information Access Evaluation,* Amsterdam, The Netherlands, 19-22 September 2011.

Olena Medelyan, Ian H. Witten, and David Milne. 2008. Topic indexing with Wikipedia. In *Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy at AAAI-08,* Chicago, Ill., 13 July 2008, pages 19–24.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of the ACM 17th Conference on Information and Knowledge Management (CIKM 2008),* Napa Valley, Cal., USA, 26–30 October 2008, pages 1046–1055.

Vivi Nastase, Michael Strube, Benjamin Börschinger, Cäcilia Zirn, and Anas Elghafari. 2010. WikiNet: A very large scale multi-lingual concept network. In *Proceedings of the 7th International Conference on Language Resources and Evaluation,* La Valetta, Malta, 17–23 May 2010.

Mark E.J. Newman. 2010. *Networks: An Introduction*. Oxford University Press, New York, N.Y.

Wayne Pullan. 2008. Approximating the maximum vertex/edge weighted clique using local search. *Journal of Heuristics*, 14(2):117–134.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *Proceedings of Coling 2010: Poster Volume,* Beijing, China, 23–27 August 2010, pages 1050–1058.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics,* Portland, Oreg., USA, 19–24 June 2011, pages 1375–1384.

Stefan Riezler and Yi Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics*, 36(3):569–582.

Helmut Schmid. 1997. Probabilistic part-of-speech tagging using decision trees. In Daniel Jones and Harold Somers, editors, *New Methods in Language Processing*, pages 154–164. London, U.K.: UCL Press.

Jianbo Shi and Jitendra Malik. 1997. Normalized cuts and image segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97),* Puerto Rico, 17–19 June 1997, pages 731–737.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics,* Portland, Oreg., USA, 19–24 June 2011, pages 793–803.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Edmonton, Alberta, Canada, 27 May –1 June 2003, pages 252–259.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A Conditional Random Field word segmenter for SIGHAN bakeoff 2005. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing,*Jeju Island, Korea, 14-15 October 2005, pages 168–171.

Denis Y. Turdakov and S. Dmitry Lizorkin. 2009. HMM expanded to multiple interleaved chains as a model for word sense disambiguation. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation,* Hong Kong, China, 3–5 December 2009.

Wolodja Wentland, Johannes Knopp, Carina Silberer, and Matthias Hartung. 2008. Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In *Proceedings of the 6th International Conference on Language Resources and Evaluation,* Marrakech, Morocco, 26 May – 1 June 2008.

Yiping Zhou. 2010. Resolving surface forms to Wikipedia topics. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 1335–1343.