Jointly Disambiguating and Clustering Concepts and Entities with Markov Logic

Angela Fahrni[1] *Michael Strube*[1]
(1) Heidelberg Institute for Theoretical Studies, Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany
Angela.Fahrni@h-its.org, Michael.Strube@h-its.org

ABSTRACT
We present a novel approach for jointly disambiguating and clustering known and unknown concepts and entities with Markov Logic. Concept and entity disambiguation is the task of identifying the correct concept or entity in a knowledge base for a single- or multi-word noun (mention) given its context. Concept and entity clustering is the task of clustering mentions so that all mentions in one cluster refer to the same concept or entity. The proposed model (1) is *global*, i.e. a group of mentions in a text is disambiguated in one single step combining various global and local features, and (2) performs disambiguation, unknown concept and entity detection and clustering *jointly*. The disambiguation is performed with respect to Wikipedia. The model is trained once on Wikipedia articles and then applied to and evaluated on different data sets originating from news papers, audio transcripts and internet sources.

KEYWORDS: Word Sense Disambiguation.

Proceedings of COLING 2012: Technical Papers, pages 815–832,
COLING 2012, Mumbai, December 2012.

1 Introduction

Recent advances in knowledge extraction from resources such as Wikipedia have allowed to create various large-scale knowledge bases and concept networks such as Yago (Suchanek et al., 2008), DBpedia (Bizer et al., 2009) or WikiNet (Nastase and Strube, 2012). To exploit the wealth of world knowledge in these resources for natural language processing tasks such as information extraction, text segmentation or summarization, words and phrases in a document first need to be linked to the relevant entries in the respective knowledge base, i.e. to be disambiguated. This problem has been tackled quite successfully by systems such as *WikipediaMiner* (Milne and Witten, 2008), which are mainly based on word sense disambiguation techniques (Agirre and Edmonds, 2006; Navigli, 2009) thus giving research on word sense disambiguation a new spin.

Concept and entity disambiguation is the task of identifying the correct concept or entity in a knowledge base for a single- or multi-word noun (*mention*) given its context.[1] In this paper we disambiguate with respect to the English Wikipedia and consider each article as a concept. One advantage of linking to Wikipedia is that the internal hyperlinks can be used as training data.

Concept disambiguation models the relation between mentions and concepts (Figure 1a). For instance, the system needs to identify if the mention *crocodile* in the first text points to AMERICAN CROCODILES (ANIMAL), to CROCODILE (LOCOMOTIVE) or to the person RENÉ LACOSTE (TENNIS PLAYER) whose nick name is *Crocodile*. We define concept disambiguation as the task of disambiguating both common nouns such as *crocodile* or *biologist* and proper nouns such as FLORIDA or STATES. While the disambiguation of common nouns is usually called *word sense disambiguation* (WSD), the disambiguation of proper nouns is also known as *entity linking*.

Although most of the work in concept disambiguation and WSD assumes that the knowledge base is complete, several studies show that many mentions have no corresponding entry in the English Wikipedia: While Zhou et al. (2010) report that between 10% and 23.5% of the mentions can not be linked to Wikipedia, Lin and Etzioni (2012) report that one third of their mentions have no corresponding entry in Wikipedia. The task of identifying mentions with no corresponding concept in the respective knowledge base is also known as recognition of NILs. In the example in Figure 1 *Aldecoa* does not refer to any entity listed in the knowledge base.

Concept clustering solves the problem of missing concepts in knowledge bases by clustering mentions within and across documents so that mentions in one cluster refer to the same concept. These clustering approaches, also known as *cross-document coreference resolution*, *sense induction* or *unsupervised word sense disambiguation* (Pedersen, 2006), do not link mentions to entries in an existing knowledge base, but cluster mentions as illustrated in Figure 1b.

We integrate the two research lines of disambiguating and clustering concepts and present a novel approach for joint disambiguation and clustering using Markov Logic (ML). Given an already existing knowledge base, mentions are linked to their corresponding entry in this knowledge base, if one exists (Figure 1a). At the same time, mentions are clustered together with other mentions that refer to the same concept, regardless of whether the referred concept exists in the knowledge base or not (Figure 1b). Figure 1c shows the joint view. In contrast most previous approaches (including systems participating at TAC (Ji et al., 2011)) use three cascaded steps: (1) Disambiguation, (2) identification of NILs, (3) clustering of NILs.

The concept selections for the different mentions (e.g. *American crocodile* and *crocodile*) are interrelated. Joint disambiguation and clustering enables us to exploit such connections:

[1] Although we use in the following the term *concepts* instead of *concepts* and *entities*, we always mean both.

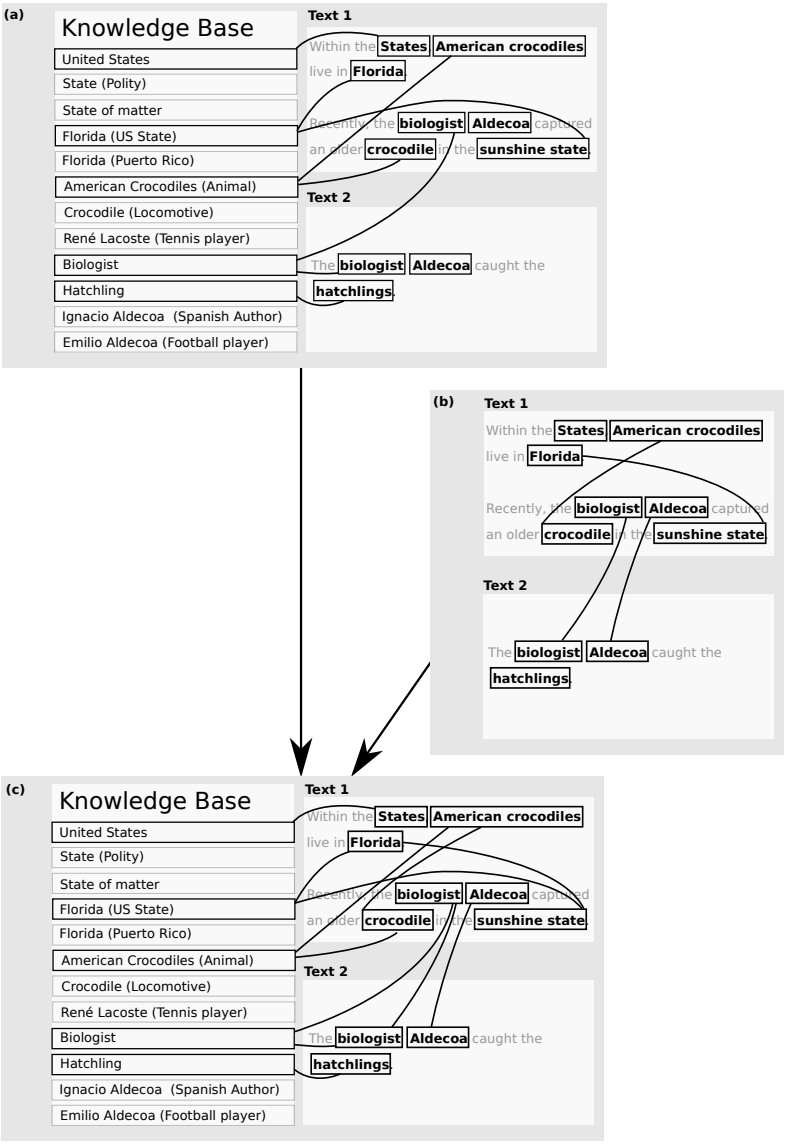

Figure 1: Joint concept disambiguation and clustering

knowledge about which mentions refer to the same concept can support disambiguation decisions. On the other hand, disambiguation influences clustering decisions. In contrast, local approaches which disambiguate mentions independently of each other (Milne and Witten, 2008; Csomai and Mihalcea, 2008) can not take advantage of such relations. Our joint approach disambiguates and clusters groups of mentions at the same time. By using Markov Logic we combine local and global features. Compared to other global models for WSD, e.g. Kulkarni et al. (2009), we do not just consider one single global feature, but combine different global features with local features and learn the weights for their combination.

Our model is trained on Wikipedia only and evaluated on ACE 2005 (annotated by Bentivogli et al. (2010)). Though we are mainly interested in the ACE data, because they provide us with annotations for proper and common nouns, we also evaluate on the TAC 2011 data which are only annotated for named entities. Nevertheless our system performs well compared to the systems participating at the TAC 2011 competition with a much smaller feature set – e.g. McNamee (2010) use 200 features – and without being trained on TAC data specifically.

The paper is organized as follows. Section 2 discusses related work, Section 3 presents our novel approach for joint disambiguation and clustering, and Section 4 presents and analyzes experiments based on ACE 2005 (Bentivogli et al., 2010) and the TAC 2011 data sets.

2 Related Work

In recent years research in monolingual and cross-lingual concept and entity disambiguation has been boosted by shared tasks such as the *Link the Wiki Track* at INEX[2], the *Cross-lingual Link Discovery Task* at NTCIR-9 (Tang et al., 2011) and the *Entity Linking Task* at TAC (McNamee and Dang, 2009; Ji et al., 2010; Ji and Grishman, 2011; Ji et al., 2011).

Dai et al. (2011) is the work that is closest to ours. In order to link gene mentions they perform entity disambiguation and recognition of the NILs at the same time using Markov Logic. In contrast to us they do not cluster mentions, but focus only on one specific type of mentions in a particular domain, namely mentions that refer to genes in a biomedical corpus.

The task of entity disambiguation, recognition of NILs and clustering has been approached in a cascaded way (Ji et al., 2011). Bunescu and Paşca (2006) first decide, if a mention refers to an entity in a knowledge base. Dredze et al. (2010) first disambiguate and then recognize the NILs. NIL recognition is often done by setting a threshold (Han and Sun, 2012). Monahan et al. (2011) interleave entity linking and clustering, but they do not approach the two tasks jointly. After disambiguation they cluster mentions. Then each cluster is assigned an entity in the knowledge base if there exists a corresponding one. Sil et al. (2012) circumvent the NIL problem by an open-database approach instead of disambiguating with respect to only one knowledge base.

Another strand of work that is similar to ours are global disambiguation approaches. While early work often uses local classifiers or rankers that select a concept for each mention independently (Csomai and Mihalcea, 2008; Milne and Witten, 2008; Dredze et al., 2010), recently, various global approaches have been proposed. Kulkarni et al. (2009) propose a method that maximizes local context-concept compatibility and global concept coherence. Fahrni et al. (2011) use a graph-based approach and select the best combination of concepts given the graph structure. Han and Sun (2012) use a generative model integrating topic coherence (one topic per document) and local context compatiblity. Ratinov et al. (2011) describe a two pass method and use

[2]http://www.inex.otago.ac.nz

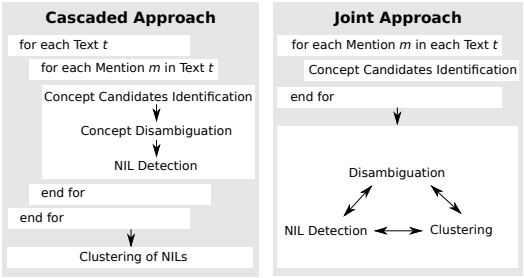

Figure 2: Cascaded approach vs. joint global approach.

the input of the first pass as input for the second one. While all these approaches use a limited number of global features, we integrate and learn the weights for various global features.

While we aim for a general domain disambiguation and clustering system that disambiguates and clusters common and proper nouns, the *Wikify!* (Csomai and Mihalcea, 2008) and *WikipediaMiner* (Milne and Witten, 2008) systems focus on the disambiguation of a few relevant keywords. Chen et al. (2012) only disambiguate person names, while Nothman et al. (2012) perform event linking.

The most prominent research line for sense induction are distributional approaches (Schütze, 1998). Pedersen (2006) gives an overview over state-of-the-art techniques. Recently, the efficency problem caused by the number of necessary comparisons has been addressed (Singh et al., 2011). While Rao et al. (2010) apply streaming clustering, Wick et al. (2012) propose a discriminative hierarchical model and partition entities into trees of latent sub-entities. None of these approaches for clustering also does concept disambiguation at the same time.

3 Approach

Markov Logic enables us to approach the task of disambiguation, recognition of unknown concepts and clustering jointly and to make use of global features. Instead of selecting for each mention – independently from earlier and later decisions – a concept, the concepts for a group of mentions are chosen at the same time.

Figure 2 contrasts the cascaded approach of disambiguation, recognition of unknown concepts and clustering with our joint global approach. As Figure 2 illustrates, first all candidate concepts for all mentions in a document are identified. Then disambiguation, recognition of NILs and clustering is performed using Markov Logic.

3.1 Markov Logic Networks

Markov Logic (ML) combines first-order logic with probabilities (Domingos and Lowd, 2009). A Markov Logic Network (MLN) consists of a set of pairs (F_i, w_i), where F_i is a first-order formula and $w_i \in \mathbb{R}$ is a weight associated with the formula F_i. It builds a template for constructing a Markov Network given a set of constants C. This Markov Network contains a binary node for each possible grounding for each predicate of the Markov Logic Network. If a ground predicate

is true the value of this binary node is 1, otherwise 0. In addition it contains one feature[3] for each ground formula. If a ground formula is true, the feature for this ground formula has the value 1, otherwise 0. The weight of the feature is given by w_i.

The probability distribution in the ground Markov Network is represented by

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right)$$

where $n_i(x)$ is the number of true groundings of F_i in x. The normalization factor Z is the partition function.

To learn the weights for the formulas and to perform MAP inference we use *thebeast*.[4] *thebeast* employs cutting plane inference (Riedel, 2008) and enables us to perform discriminative training using a perceptron.

3.2 Disambiguation and Clustering with Markov Logic

The backbone of our model is the definition of how disambiguation, recognition of NILs and clustering interact. To model these relations we use hard constraints. In the following we will first describe these constraints, before we explain the features in the next section.

Table 1 shows all predicates and formulas used. Each formula is associated with a positive or negative weight. While the weight – except for hard constraints – is learned from training data, the polarity of the weights is set manually. In the following we indicate the direction by the + or − in front of each formula. Formulas with negative weights provide evidence for recognizing NILs. For some formulas the final weight consists of a learned weight w multiplied by a score s (e.g. prior probability). In these cases the final weight for a formula does not just depend on the respective formula, but also on the instantiation, e.g. a specific mention and candidate concept. We indicate such combined weights by the term $w \cdot s$, while w refers to cases where the formula is exclusively weighed by the learned weight. M denotes all mentions and C_m refers to all candidate concepts of a mention m.

Disambiguation and clustering are two different perspectives on the problem of lexical ambiguities. While in concept disambiguation the focus lies on the relation between mentions and concepts, clustering deals with relations between mentions. The tasks of disambiguation and recognition of unknown concepts are interrelated, as both tasks look at mention–concept relations. However, while in concept disambiguation the question is to which concept a mention refers to given its context, the task of recognizing unknown concepts is to determine, if such a concept relation exists for a given mention at all.

To approach disambiguation, recognition and clustering of NILs with ML we define a hidden predicate for each relation we are interested in. The predicate *hasConcept(Mention, Concept)* models the relation between mentions and concepts in the knowledge base (Table 1, p1). To ensure that each mention refers to at most one concept a hard cardinality constraint is defined: for each mention the predicate *hasConcept* is true at most once. This constraint allows us to jointly disambiguate and recognize NILs (Table 1, f1).

[3] Note that *feature* is used differently in this section than in the rest of the paper.
[4] `http://code.google.com/p/thebeast`.

To model if two mentions refer to the same concept, the predicate *hasSameConcept(Mention, Mention)* is used (Table 1, p2). It is true for all mention pairs that refer to the same concept, regardless whether the referred concept exists in the knowledge base or not. This clustering relation is *transitive* and *symmetric* (Table 1, f2, f3).

In order to perform joint disambiguation and clustering it needs to be defined how the mention-concept relation (disambiguation, recognition of NILs) and the clustering relation are interrelated (Table 1, f4, f5). Given that two mentions refer to the same concept in the knowledge base they belong to the same cluster (f5). On the other hand, if two mentions are part of the same cluster and one of them refers to a concept in the knowledge base, the other mention in the cluster has to refer to the same concept (f4). Note, two mentions can also be in the same cluster without referring to a concept in the knowledge base.

3.3 Features

All features have a corresponding predicate which is part of at least one formula. In the following we focus on the features.

3.3.1 Local Features

Local features involve one single mention and its candidate concepts.

Prior probability (p3, f7) The prior probability is defined as the probability that a mention m refers to a concept c. To estimate this probability all internal hyperlinks are extracted from the English Wikipedia dump. For each linked mention (m) it is counted, how many times it links to a particular Wikipedia page ($count_{m,c}$), i.e. concept. This count is normalized by the number of times mention m is linked to Wikipedia pages ($count_m$):

$$p(c|m) = \frac{count_{m,c}}{count_m}$$

Relatedness (p4, f8, f11) This feature reflects the average pairwise relatedness of a candidate concept for a mention to the context and is calculated in the same way as proposed by Milne and Witten (2008). The pairwise relatedness is calculated via

$$rel(c_1, c_2) = \frac{log(max(|C_1|, |C_2|)) - log(|C_1 \cap C_2|)}{log(|W|) - log(min(|C_1|, |C_2|)}$$

where c_1 and c_2 are two concepts, C_1 and C_2 denotes the articles in Wikipedia that link to the articles c_1 and c_2 respectively, and W is the total number of articles in Wikipedia. The more a candidate is related to the context, the more likely it is that a mention refers to it (f8). If a candidate concept for a mention is not at all related to the context, i.e. the average relatedness is zero, this is a negative indicator for a candidate (f11).

Local context similarity (p5, f9) The local context similarity measures how similar the current local context (K_m) – consisting of seven words before and after the mention – is to the local contexts for that concept in Wikipedia. For each mention in the English Wikipedia that is linked to a certain Wikipedia page c we extract the surrounding words (T_c) using the same context definition as above. We then calculate the local context similarity ($sim(c, m)$) for a candidate concept c of a mention m via

$$sim(c,m) = \frac{1}{|K_m|} \sum_{k \in K_m} s(k, T_c)$$

where the first term is used for normalization and $s(k, T_c)$ denotes the frequency of k in T_c divided by the number of times k appears in the context of all concepts in Wikipedia.[5]

String edit distance (p6, f10) This feature accounts for the difference between the mention string used in the text (m) and the preferred name (p) for a candidate concept of m. We assume that the Wikipedia article title and the titles of its redirects are preferred names for a concept (P). To measure the distance between preferred names and the mention in the text we calculate the edit distance[6] ($dist_{m,p}$) and normalize it by the length of the longer string:

$$sim_{m,p} = \frac{dist_{m,p}}{max(|m|, |p|)}$$

If there exists more than one preferred term for a concept, we take the maximum. This feature indicates a negative relation between a candidate concept and a mention. The more distant a preferred name is from a mention, the less likely it is that the mention refers to this concept.

3.3.2 Global Features

In contrast to local features, global features involve more than one mention. From a disambiguation perspective these features define, which mentions are disambiguated jointly.

Shared lemma (p7, f12) The one sense per discourse assumption states that one mention string is used to refer to one sense, i.e. in our case to one concept, in one discourse (Gale et al., 1992). For each document we extract all mentions with the same lemma and the inverse distance in sentences between the two. The bigger the inverse distance is, the closer the two mentions are to each other and the more likely it is that they refer to the same concept.

Head match (p8,f6) The one concept per discourse assumption often applies to mentions that are in a substring relation and share the same syntactic head lemma. We extract all these pairs including the inverse distance between the respective mentions.

Acronyms (p8,f6) In texts, especially in news paper texts, acronyms are often introduced by the following pattern: full name (acronym). We extract all these mention pairs, whereas one mention is the full name and the other one the acronym.[7]

Cross-document n-gram feature (p9,f13) In contrast to the previous features this one is a cross-document feature. The assumption is that we work with a document collection. We extract all mention pairs with the same lemma but coming from two different documents. For each of these mentions we extract all n-grams that include the respective mention and that consist of nouns and adjectives. If the two mentions share at least one of these n-grams, we consider them as referring to the same concept and add as score the number of shared n-grams.

[5] We take its logarithm.

[6] We use the Lingpipe implementation (`http://alias-i.com/lingpipe/`).

[7] In our Wikipedia training data, acronyms are relatively rare. Hence it is difficult to learn a weight for the acronym feature. As it is similar to the head match feature, we use the same predicate and weight for the two features.

Predicates

Hidden predicates	
p1	$hasConcept(m, c)$
p2	$hasSameConcept(m, n)$
Predicates realizing Wikipedia Miner features	
p3	$hasPriorProbability(m, c, s)$
p4	$hasRelatedness(m, c, s)$
Additional predicates involving one mention and one entity	
p5	$hasContextSimilarity(m, c, s)$
p6	$hasStringDistance(m, c, s)$
Predicates involving two mentions (intradocument)	
p7	$isSubStringHeadMatch(m, n, s)$
p8	$haveSameLemma(m, n, s)$
Predicates involving two mentions (cross-document)	
p9	$shareNgram(m, n, s)$

Formulas

Hard constraints			
f1			$\forall m \in M : \lvert\{c \in C : hasConcept(m, c)\}\rvert \leq 1$
f2			$\forall m, n \in M : m \neq n \wedge hasSameConcept(m, n) \rightarrow hasSameConcept(n, m)$
f3			$\forall m, n, l \in M : m \neq n \wedge m \neq l \wedge n \neq l$ $\wedge hasSameConcept(m, n) \wedge hasSameConcept(n, l) \rightarrow hasSameConcept(m, l)$
f4			$\forall m, n \in M : m \neq n \wedge hasSameConcept(m, n) \wedge hasConcept(m, c)$ $\rightarrow hasConcept(n, c)$
f5			$\forall m, n \in M : m \neq n \wedge m \neq n \wedge hasConcept(m, c) \wedge hasConcept(n, c)$ $\rightarrow hasSameConcept(m, n)$
Formulas with learned weights			
f6	$+$	$(w \cdot s)$	$\forall m, n \in M\ \forall c \in C_m : m \neq n \wedge isSubStringHeadMatch(m, n, s)$ $\rightarrow hasConcept(m, c) \wedge hasConcept(n, c)$
f7	$+$	$(w \cdot s)$	$\forall m \in M\ \forall c \in C_m : hasPriorProbability(m, c, s) \rightarrow hasConcept(m, c)$
f8	$+$	$(w \cdot s)$	$\forall m \in M\ \forall c \in C_m : hasRelatedness(m, c, s) \rightarrow hasConcept(m, c)$
f9	$+$	$(w \cdot s)$	$\forall m \in M\ \forall c \in C_m : hasContextSimilarity(m, c, s) \rightarrow hasConcept(m, c)$
f10	$-$	$(w \cdot s)$	$\forall m \in M\ \forall c \in C_m : hasStringDistance(m, c, s) \rightarrow hasConcept(m, c)$
f11	$-$	(w)	$\forall m \in M\ \forall c \in C_m : hasRelatedness(m, c, s) \wedge s = 0 \rightarrow hasConcept(m, c)$
f12	$+$	$(w \cdot s)$	$\forall m, n \in M : m \neq n \wedge hasSameString(m, n, s) \rightarrow hasSameConcept(m, n)$
f13	$+$	$(w \cdot s)$	$\forall m, n \in M : m \neq n \wedge shareNgram(m, n, s) \rightarrow hasSameConcept(m, n)$

Table 1: Predicates and formulas used for disambiguation and clustering (m, n, l represent mentions, M sets of mentions, c concepts and entities, C sets of concepts and entities, and s scores)

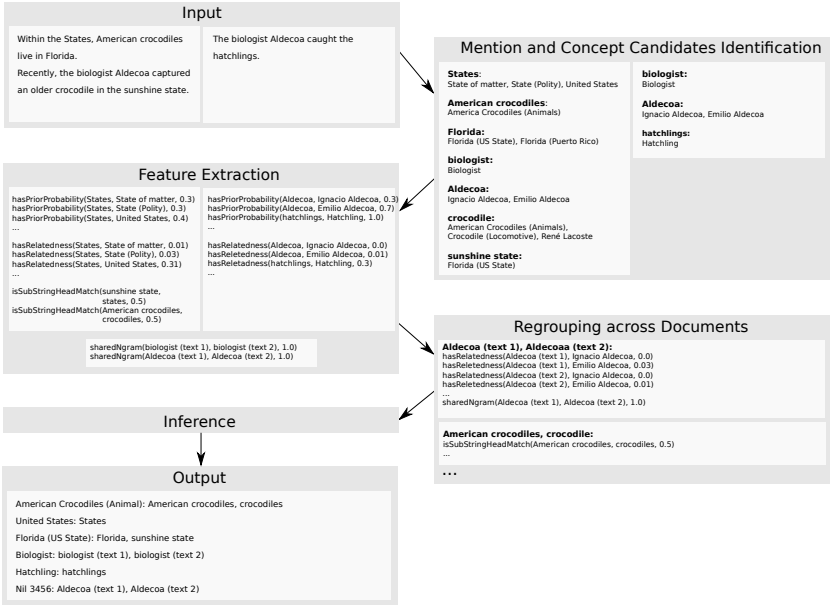

Figure 3: Example for the whole process.

3.4 Illustration of Our Approach

Given several documents – in Figure 3 just two – we first detect mentions in these documents by identifying noun phrases. If a mention is in our lexicon (see Section 4), we obtain all candidate concepts from there. Otherwise we just keep the mention, if it is at most of a length of four tokens and starts and ends with a noun. We keep mentions with no candidate concepts for two reasons: (1) It can happen that during disambiguation and clustering a mention is linked to a concept even if it does not have a candidate concept. (2) As we also want to cluster mentions with no candidate concept, they need to be kept in the network. In the next step we extract all features, we use for disambiguation and clustering (see Section 3.2). As features can cross document boundaries, we then regroup the mentions into pseudo documents, which are given to the inference module. One pseudo document contains all mentions that are linked by global features. The output is shown at the bottom of Figure 3.

4 Experiments

This section describes experiments on two different data sets. All experiments, including the ones for the baseline systems such as *WikipediaMiner*, are based on the same English Wikipedia dump[8], the same lexicon, which includes all anchor texts that occur more than two times with a certain concept (Milne and Witten, 2008),[9] and the same preprocessing. This way we ensure

[8]We use the dump from January 4th, 2012.

[9]We use a query expansion technique and also consider redirects and article titles and retrieve the candidate for the closest anchor.

Dataset	No. of Documents	No. of Mentions	in KB	NILs	Ave. Amb.
WP Training	500	46,810	43,547	3,263	2.18
WP Dev	100	7,197	6,610	587	2.11
ACE 2005	597	29,300	27,184	2,116	5.72
TAC 2011	2162	2250	1124	1126	4.51

Table 2: Datasets: Statistics

that differences in the results are caused exclusively by algorithm and features.

4.1 Data Sets

Disambiguating with respect to Wikipedia has the advantage that training data can be derived from the internal hyperlinks in Wikipedia automatically without manual annotation.

While training and development is done exclusively on Wikipedia (WP Training, WP Dev), we evaluate our approach and compare it to previous work using two data sets containing texts from different sources such as news papers, audio transcription records and the internet (ACE 2005, TAC 2011). Table 2 summarizes some statistics for each data set, namely the number of documents and mentions to disambiguate, the number of mentions with a corresponding concept in the knowledge base (in KB), the number of NILs and the average ambiguity.

4.1.1 Training and Development Data

For training and development we use featured articles from the English Wikipedia data (featured articles are supposed to be of high quality). We randomly select articles among those articles and consider all internal hyperlinks that point to an existing article as an concept-annotated mention. In Wikipedia only the first occurrence of a concept in an article is linked to the respective page. Since our aim is to disambiguate all occurrences we re-wikify the articles: all mentions, that – according to our lexicon – can refer to a linked article, are linked automatically to the respective concept. For training we collect for each annotated mention all candidate concepts from our lexicon. For obtaining NILs we randomly remove some of the concepts from the annotations and the lexicon. The development data set is processed in the same way.

4.1.2 Testing Data

The English part of the *ACE 2005* data set has been manually annotated with links to Wikipedia by Bentivogli et al. (2010). ACE 2005 consists of 597 texts from newswire reports, broadcast news, internet sources and transcribed audio data. Both common and proper nouns that are part of a coreference chain are annotated with one or more links to the English Wikipedia or as NILs. Some of the mentions are annotated with more than one link. We consider a mention as correctly disambiguated, if one of the annotated links is identified.

The English TAC dataset from 2011 consists of 2,250 queries and focuses on named entities such as persons, organizations and locations. A query consists of a query term, i.e. a name for a named entity, and a document, in which the query term appears. The documents are newspaper and web texts. Given our approach we disambiguate the whole document and not just the query terms. In contrast to ACE 2005 the NILs are not just annotated as NILs but also clustered, which allows us to evaluate the entity clustering performance in a direct way and not just its influence on the disambiguation performance as on the ACE data.

4.2 Experiments

4.2.1 Baselines, Other Systems, Upper Bounds

To evaluate our approach we compare it to different baseline systems. For all systems and baselines we cluster the remaining unclustered NILs in a postprocessing step using a string match heuristic, which is a hard-to-beat baseline for this task (Ji et al., 2011). For the upper bound we assume that the clustering is perfect given the disambiguation results.

Upper bound I (UB): The first upper bound shows the performance which can maximally be reached using our lexicon. Each mention is considered as correctly disambiguated, if the correct concept is among the candidates for that mention given our lexicon. If a mention is a NIL according to our gold standard, we also consider it as correct.

Upper bound II (UBD): The second upper bound considers all candidate concepts for all mentions in a document as candidate concepts for a mention. If the correct concept is among these candidates, it is considered as correct. This is the upper bound for our final ML system.

First concept baseline (First concept): In WSD the first sense baseline is known as hard to beat as the distribution of the concepts given a mention obeys Zipf's law. The first concept baseline only makes use of the prior probability of a concept given a mention and always selects the one with the highest prior probability.

WikipediaMiner: WikipediaMiner (Milne and Witten, 2008) is a state-of-the-art system which is freely available. We use version 1.1, extract all the necessary information from the 2012 Wikipedia dump and train it on the same training data that we use to learn the weights for our ML systems.

SVM Rank I (SR I): A common approach for named entity disambiguation is to use a ranker to rank the candidate concepts for each mention and then select the highest ranked concept for each mention. We use SVM^{Rank} (Joachims, 2002) trained on our Wikipedia training data using the same features as for *MLN Dis.* (see below).

SVM Rank II (SR II): This system also uses SVM^{Rank} (Joachims, 2002) and the same features as for *ML Dis.+NILs* (see below).

SVM Rank NIL Classifier II (SRC II): This system uses the *SVM Rank II* system to obtain for each mention the highest ranked concept. Then we apply a classifier to decide for each mention-concept pair, if it is a valid mapping or if the mention is a NIL. We use decision trees as a classifier (Witten and Frank, 2005) and the same features as for *ML Dis.+NILs*.

Other systems: For the TAC 2011 data we also add the best (**Best system**) (Monahan et al., 2011) and median (**Median system**) performance of all participating systems in the English entity linking task at TAC 2011.

4.2.2 Our Systems

ML Dis.: ML system using predicates p1, p3, p4, the local formulas f7, f8 and the global constraint f1. This system uses only information that is also used by WikipediaMiner and do not recognize NILs as no negative information is integrated.

MLN Dis.+NILs: ML system using predicates p1, p3-p6, the local formulas f7-f11 and the global constraint f1. While *ML Dis.* assigns each mention a concept, this system performs concept disambiguation and recognition of NILs jointly.

ACE 2005							
	Non NILs			NILs			
	P	R	F	P	R	F	Acc
UB	90.1	87.3	88.7	71.4	100.0	83.3	88.2
UBD	95.8	93.3	94.5	75.0	100.0	85.7	93.8
First Concept	68.3	69.4	68.8	49.9	39.5	44.1	67.2
WikipediaMiner	86.5	59.1	70.2	16.9	85.8	28.3	61.0
SR I	69.1	70.2	69.7	49.9	39.5	44.1	68.0
SR II	69.7	70.8	70.2	49.9	39.5	44.1	68.5
SRC II	79.7	57.8	67.0	19.0	86.0	31.1	59.8
ML Dis.	71.2	72.4	71.8	49.9	39.5	44.1	70.0
ML Dis.+NILs	79.0	74.2	76.5	36.0	63.9	46.1	73.5
ML Dis.+NILs+Clust.	78.0	75.6	76.7	41.1	57.4	47.9	74.3

Table 3: Evaluation on ACE 2005 data

ML Dis.+NILs+Clust.: Joint ML system that performs disambiguation, recognition of NILs and clustering jointly. While we use all predicates and formulas for the TAC data, we do not consider predicate p9, formula f13 on the ACE data.

4.2.3 Results

Table 3 and Table 4 show the results on the *ACE 2005* and *English TAC 2011* data set respectively. We report precision (P), recall (R) and F-measure (F) for the non NILs and the NILs as well as the overall accuracy (Acc), also known as micro-average. In addition we also present the B^3 precision, recall and F-measure for the *TAC data sets* in order to evaluate the entity clustering performance.[10] Significance is calculated for the overall accuracy using a paired t-test.

4.2.4 Discussion of the Results

The full system, which does joint disambiguation, recognition of NILs and clustering (*ML Dis.+NILs+Clust.*), significantly outperforms the other systems in the two tables – except the best performing system at TAC 2011 (*Best system*) – with $p < 0.01$. Bryl et al. (2010) report on the ACE data an F-Measure of 71.5 for non-NILs, but these results are not comparable as they use gold mentions instead of system mentions and consider a mention as correctly disambiguated only if it links to the first mentioned Wikipedia article in the gold standard. Ratinov et al. (2011) – another state-of-the-art system – report an accuracy of 78.8 and a B^3 F-Measure of 76.2 on the TAC 2011 data set (Ratinov and Roth, 2011).

The system *ML Dis.* is close to *WikipediaMiner*. It uses only positive evidence, i.e. the two described *WikipediaMiner* features, and links each mention to a concept in Wikipedia. No NILs are identified. Hence the results for the NILs on the ACE data set are the same as for the first concept baseline (*First Concept*), the ranking systems (*SR I* and *SR II*), which also assign each mention a concept, and *ML Dis.*. On the TAC 2011 data set the results for the NILs differ between the *First Concept* baseline, the ranking systems (*SR I, SR II*) and *ML Dis.*. The difference comes from the fact that the TAC knowledge base is not identifical with the knowledge base we use for disambiguation. If the system links a mention to an entry in our knowledge base, which is not part of the TAC knowledge base, it is considered as NIL. The system *ML Dis.+NILs* performs disambiguation and recognition of NILs jointly. Compared to

[10] We use the offical TAC scoring scripts.

TAC EN Test 2011										
	Non NILs			NILs						
	P	**R**	**F**	**P**	**R**	**F**	**Acc**	B^3 **P**	B^3 **R**	B^3 **F**
UB	100.0	75.0	85.7	80.0	100.0	88.9	87.5	87.5	87.2	87.4
UBD	100.0	95.2	97.5	95.4	100.0	97.7	97.6	97.6	97.4	97.5
First Concept	61.8	54.2	57.7	76.5	85.9	80.9	70.0	65.4	69.6	67.4
WikipediaMiner	86.1	55.1	67.2	70.0	95.2	80.7	75.2	70.7	73.7	72.2
SR I	72.8	66.5	69.5	81.5	88.5	84.8	77.5	73.7	76.4	75.0
SR II	73.2	66.9	69.9	81.2	88.2	84.5	77.6	73.7	76.5	75.1
SRC II	87.7	59.1	70.6	72.3	95.8	82.4	77.5	73.3	74.6	73.9
Best System										84.6
Median System										71.6
ML Dis.	71.4	65.5	68.3	81.4	88.1	84.6	76.8	72.9	75.7	74.3
ML Dis.+NILs	79.5	64.1	70.9	77.5	92.5	84.3	78.3	74.2	76.6	75.4
ML Dis.+NILs+Clust.	80.3	74.5	77.3	85.1	91.3	88.1	82.9	79.2	81.1	80.1

Table 4: Evaluation on TAC 2011

the two step process (*SRC II*), which uses the same features, but performs ranking and the classification of NILs in a cascaded way, the performance of the joint approach (*ML Dis.+NILs*) is higher. As the differences between the systems *ML Dis.+NILs* and *ML Dis.+NILs+Clust.* show, addressing clustering and disambiguation jointly improves the results even further. The improvement mainly comes from two different cases: (1) Mentions with no candidate concepts, which are recognized as NILs in *ML Dis.+NILs* are correctly disambiguated and clustered by *ML Dis.+NILs+Clust*. While for example *ML Dis.+NILs* recognized *Marinello* in "We pretty much know that Marinello, while on the board, has arranged to get future money" as a NIL, *ML Dis.+NILs+Clust.* links it to the correct entry in the knowledge base by also taking into account other occurrences of *Marinello* such as "because the fact that Randy Bauer is already talking about *Beatriz Marinello*". (2) Wrongly disambiguated mentions are – thanks to discourse knowledge – correctly disambiguated. This especially applies to occurrences of common nouns such as *region* or *friends*. Whereas the system *ML Dis.+NILs* wrongly disambiguated *friends* as the TV series, *ML Dis.+NILs+Clust.* correctly links it to the entry on friendship by taking into account other occurrences of *friends* in the text.

5 Conclusions

This paper presents a new approach for joint disambiguation, NIL recognition and clustering using Markov Logic. Our approach significantly outperforms all baseline systems and shows state-of-the-art performance. To our knowledge this is the first approach for joint disambiguation and clustering of concepts and entities. At the moment, we tested on a relatively small data set. For future work, we will work on scalability and on more linguistically informed features.

Acknowledgments. We would like to thank Mathias Niepert for his advice on Markov Logic and *thebeast*. This work has been partially funded by the European Commission through the CoSyne project FP7-ICT-4-248531 and the Klaus Tschira Foundation.

References

Agirre, E. and Edmonds, P. G., editors (2006). *Word Sense Disambiguation: Algorithms and Applications*. Springer, Heidelberg, Germany.

Bentivogli, L., Forner, P., Giuliano, C., Marchetti, A., Pianta, E., and Tymoshenko, K. (2010). Extending English ACE 2005 corpus annotation with ground-truth links to Wikipedia. In *Proceedings of the 2nd Workshop on The People's Web: Colloboratively Constructed Semantic Resources,* Beijing, China, 28 August 2010, pages 19–27.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). DBPedia – A crystallization point for the Web of data. *Journal of Web Semantics*, 7:154–165.

Bryl, V., Giuliano, C., Serafini, L., and Tymoshenko, K. (2010). Supporting natural language processing with background knowledge: Coreference resolution case. In *Proceedings of the 9th International Semantic Web Conference, Revised Selected Papers, Part I,* Shanghai, China, 7-11 November 2010, pages 80–95.

Bunescu, R. and Paşca, M. (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.

Chen, L., Feng, Y., Zou, L., and Zhao, D. (2012). Explore person specific evidence in web person name disambiguation. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 832–842.

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

Dai, H.-J., Tsai, R. T.-H., and Hsu, W.-L. (2011). Entity disambiguation using a Markov-Logic network. In *Proceedings of the 5th International Joint Conference on Natural Language Processing,* Chiang Mai, Thailand, 8–13 November 2011, pages 846–855.

Domingos, P. and Lowd, D. (2009). *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan Claypool Publishers.

Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 277–285.

Fahrni, A., Nastase, V., and Strube, M. (2011). HITS' graph-based system at the NTCIR-9 cross-lingual link discovery task. In *Proceedings of the 9th NTCIR Workshop Meeting,* Tokyo, Japan, 6-9 December 2011, pages 473–480.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*.

Han, X. and Sun, L. (2012). An entity-topic model for entity linking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* Jeju Island, Korea, 8–14 July 2012, pages 105–115.

Ji, H. and Grishman, R. (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.

Ji, H., Grishman, R., and Dang, H. (2011). Overview of the TAC 2011 knowledge base population track. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 14–15 November 2011.

Ji, H., Grishman, R., Dang, H., Griffitt, K., and Ellis, J. (2010). Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 15–16 November 2010.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining,* Edmonton, Canada, 23–26 July 2002, pages 133–142.

Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (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.

Lin, T. and Etzioni, O. (2012). Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX),* Montréal, Québec, Canada, 7–8 June 2012, pages 84–88.

McNamee, P. (2010). HLTCOE efforts in entity linking at TAC 2010. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 15–16 November 2010.

McNamee, P. and Dang, H. T. (2009). Overview of the TAC 2009 knowledge base population track. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 16–18 November 2009.

Milne, D. and Witten, I. H. (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.

Monahan, S., Lehmann, J., Nyberg, T., Plymale, J., and Jung, A. (2011). Cross-lingual cross-document coreference with entity linking. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 14–15 November 2011.

Nastase, V. and Strube, M. (2012). Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Nothman, J., Honnibal, M., Hachey, B., and Curran, J. R. (2012). Event linking: Grounding event reference in a news archive. In *Proceedings of the ACL 2012 Conference Short Papers,* Jeju Island, Korea, USA, 8–14 July 2012, pages 228–232.

Pedersen, T. (2006). Unsupervised corpus-based methods for WSD. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation: Algorithms and Applications*, pages 133–166. Springer, Heidelberg, Germany.

Rao, D., McNamee, P., and Dredze, M. (2010). Streaming cross document entity coreference resolution. In *Proceedings of Coling 2010: Poster Volume,* Beijing, China, 23–27 August 2010, pages 1050–1058.

Ratinov, L. and Roth, D. (2011). GLOW TAC-KBP2011 entity linking system. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 14–15 November 2011.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (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.

Riedel, S. (2008). Improving the accuracy and efficiency of MAP inference for Markov logic. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence,* Helsinki, Finland, 9–12 July 2008, pages 468–475.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Sil, A., Cronin, E., Nie, P., Yang, Y., Popescu, A.-M., and Yates, A. (2012). Linking named entities to any database. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 116–127.

Singh, S., Subramanya, A., Pereira, F., and McCallum, A. (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.

Suchanek, F., Kasneci, G., and Weikum, G. (2008). YAGO: A large ontology from Wikipedia and WordNet. *Elsevier Journal of Web Semantics*, 6(3):203–217.

Tang, L.-X., Geva, S., Trotman, A., Xu, Y., and Itakura, K. (2011). Overview of the NTCIR-9 crosslink task: Cross-lingual link discovery. In *Proceedings of the 9th NTCIR Workshop Meeting,* Tokyo, Japan, 6-9 December 2011, pages 437–463.

Wick, M., Singh, S., and McCallum, A. (2012). A discriminative hierarchical model for fast coreference at large scale. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* Jeju Island, Korea, 8–14 July 2012, pages 379–388.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, Cal., 2nd edition.

Zhou, Y., Nie, L., Rouhani-Kalleh, O., Vasile, F., and Gaffney, S. (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.