Joint Entity Recognition and Linking in Technical Domains Using Undirected Probabilistic Graphical Models*

Hendrik ter Horst, Matthias Hartung and Philipp Cimiano

Cognitive Interaction Technology Cluster of Excellence (CITEC)
Semantic Computing Group
Bielefeld University
{hterhors, mhartung, cimiano}@techfak.uni-bielefeld.de

Abstract. The problems of recognizing mentions of entities in texts and linking them to unique knowledge base identifiers have received considerable attention in recent years. In this paper we present a probabilistic system based on undirected graphical models that jointly addresses both the entity recognition and the linking task. Our framework considers the span of mentions of entities as well as the corresponding knowledge base identifier as random variables and models the joint assignment using a factorized distribution. We show that our approach can be easily applied to different technical domains by merely exchanging the underlying ontology. On the task of recognizing and linking disease names, we show that our approach outperforms the state-of-the-art systems *DNorm* and *TaggerOne*, as well as two strong lexicon-based baselines. On the task of recognizing and linking chemical names, our system achieves comparable performance to the state-of-the-art.

Keywords: Joint Entity Recognition and Linking; Undirected Probabilistic Graphical Models; Diseases; Chemicals

1 Introduction

In light of the current proliferation of openly accessible textual data and structured symbolic knowledge in the LOD cloud[1], a versatile approach to the representation of text meaning relies on linking mentions in a text to entities, relations or classes defined in a reference knowledge base such as DBpedia[2] or MeSH[3]. Being coined as named entity disambiguation, entity linking, or wikification, this task has received considerable attention in recent years [7, 16, 15].

* This is a pre-print version of an article published by Springer in LNAI 10318: https://link.springer.com/chapter/10.1007%2F978-3-319-59888-8_15

[1] http://lod-cloud.net/

[2] http://wiki.dbpedia.org

[3] Medical Subject Headings: https://www.nlm.nih.gov/mesh

As a subtask in machine reading, i.e., automatically transforming unstructured natural language text into structured knowledge [19], entity linking facilitates various applications such as entity-centric search or predictive analytics in knowledge graphs. In these tasks, it is advisable to search for the entities involved at the level of unique knowledge base identifiers rather than surface forms mentioned in the text, as the latter are ubiquitously subject to variation (e.g., spelling variants, semantic paraphrases, or abbreviations). Thus, entities at the concept level cannot be reliable retrieved or extracted from text using exact string match techniques.

Prior to linking the surface mentions to their respective concepts, named entity recognition [17] is required in order to identify all sequences of tokens in the input sentence that denote an entity of a particular type (e.g., diseases or chemicals). Until recently, named entity recognition and entity linking have been mostly performed as separate tasks in pipeline architectures ([7, 20], inter alia).

In this paper, we present *J-Link*, a versatile approach to *joint* entity recognition and linking that can be easily applied to different technical domains by exchanging the underlying knowledge base and training data. The approach exploits undirected probabilistic graphical models (factor graphs, in particular) and Markov Chain Monte Carlo methods for inference. Parameter updates are computed using SampleRank [24].

We train and evaluate the system in two experiments focusing on joint entity recognition and linking of diseases and chemical compounds, respectively. In both tasks, the BioCreative V CDR data [23] is used for training and testing. We apply the same model to both problems, only exchanging the underlying reference knowledge base. With an F_1 score of 85.9 in disease linking, we outperform the state-of-the-art systems *DNorm* [12] and *TaggerOne* [11]; in chemical compounds linking, our system achieves an F_1 score of 86.6, which is comparable to the state-of-the-art. Thus, J-Link provides high performance on both domains without major need of manual adaptation or system tuning.

2 Related Work

Entity linking approaches have mostly relied on three main sources of information: *Local models* investigate the textual context of a surface entity mention ([15], inter alia), *global models* aim at collective linking of all entities within the same document ([20], inter alia), and *graph-based models* focus on the relation between surface mentions and entity candidates ([16], inter alia).

More recently, these sources have been combined in probabilistic graphical models. The approach by Hakimov et al. [5] incorporates textual and graph-based features in a factor graph model in order to capture compatibilities of pairs of mentions and entities within the same document. Their results show that entity co-occurrences and mention-entity pairs provide complementary information to the model. Based on the same sources of information, Ganea et al. [4] train a Markov network for the entity linking task, using approximate MAP inference by

belief propagation. Both Ganea et al. and Hakimov et al. perform entity linking in isolation by relying on gold annotations for the recognition problem.

Probabilistic graphical models can be used to couple the tasks of named entity recognition and entity linking in joint models such that mutual dependencies between both problems are exploited. This avenue has recently been explored by Durrett and Klein [2], Luo et al. [14] and Nguyen et al. [18]. Consistently, these approaches extend conditional random fields (CRF; [10]) which constitute the state-of-the-art in named entity recognition. By extending linear-chain CRFs to tree-shaped factor graphs based on syntactic dependency relations between variables, non-local features considering entity-entity pairs or entity-level priors can be incoporated as well [3]. In our work presented here, we adopt an even more flexible model structure which is sufficiently versatile to encode non-local information, while it does not require dependency parsing.

In contrast to the latter approaches which all use Wikipedia as reference knowledge base, there are several domain-specific approaches to entity linking. We focus our discussion on the biomedical domain and disease/chemical recognition and linking, as this is our application scenario in this paper. The DNorm system [12] relies on a learning-to-rank approach in order to induce similarities between disease mentions and concept names directly from training data. However, the system does not include any information about coherence between different entities within the same text. In contrast to DNorm, TaggerOne [11] performs entity recognition and linking simultaneously, using a combination of semi-Markovian sequence labeling (for the recognition problem) and supervised semantic indexing (for the linking problem). These components do not share any parameters, i.e., possible dependencies between the individual problems are not captured in the model. The system by Lee et al. [13], combining disease recognition and linking in a sequential pipeline architecture, obtained the best performance at the BioCreative V Shared Task on disease linking [23]. However, their approach is specifically tailored to the domain as it strongly capitalizes on strategies for expanding the reference knowledge base, which is not our focus in this work. Instead, we aim at a more general model for joint entity recognition and linking that can be flexibly adapted to knowledge bases from various domains. In that respect, our work follows similar goals as the AGDISTIS framework [22], which performs entity linking that is agnostic of the underlying knowledge base, without considering the recognition problem, though.

3 Method

We frame the entity recognition and linking tasks as a joint inference problem in an undirected probabilistic graphical model framework. In such a model, a factor graph representation is used to decompose a joint probability distribution over observed and hidden random variables. In the following, we (i) describe the notion of factor graphs, (ii) show how we use them to represent the problem domain for joint entity recognition and linking, and (iii) how we perform inference over factor graphs using Markov Chain Monte Carlo sampling (Sections 3.1–

3.3). In Section 3.4, we describe how the parameters of our model are optimized using SampleRank. Section 3.5 presents the methods used in order to retrieve candidate concepts from a reference knowledge base. The features of our model are described in Section 3.6.

3.1 Factor Graphs

Following Kschischang et al. [9] and Hakimov et al. [5], we define a factor graph $\mathcal{G}$ as a bipartite graph that consists of variables V and factors Ψ. Variables can further be divided into *observed* variables $\boldsymbol{x}$ and *hidden* variables $\boldsymbol{y}$. A factor Ψ_i connects subsets of observed variables $\boldsymbol{x}_i$ and hidden variables $\boldsymbol{y}_i$. Each factor computes a scalar score based on the exponential of the scalar product of a feature vector $f_i(\boldsymbol{x}_i, \boldsymbol{y}_i)$ to be determined from the corresponding subset of variables and a set of parameters θ_i: $\Psi_i = e^{f_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \theta_i}$. Based on these definitions, the inference problem in factor graphs, i.e., computing the posterior distribution of the hidden variables given the observed ones, can be formulated in terms of the product of the individual factors:

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{1}{Z(\boldsymbol{x})} \prod_{\Psi_i \in \mathcal{G}} e^{\Psi_i} = \frac{1}{Z(\boldsymbol{x})} \prod_{\Psi_i \in \mathcal{G}} e^{f_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \cdot \theta_i} \tag{1}$$

where $Z(\boldsymbol{x})$ is the normalization function.

For a given set of observed variables, we generate a factor graph automatically making use of factor templates $\mathcal{T}$. Each template $T_j \in \mathcal{T}$ defines (i) the subsets of observed and hidden variables $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ for which it can generate factors and (ii) a function $f_j(\boldsymbol{x}_j, \boldsymbol{y}_j)$ to generate features for these variables. All factors generated by a given template T_j share the same parameters θ_j. With this definition, we can reformulate the conditional probability from Equation (1) as follows:

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{1}{Z(\boldsymbol{x})} \prod_{T_j \in \mathcal{T}} \prod_{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in T_j} e^{f_j(\boldsymbol{x}_j, \boldsymbol{y}_j) \cdot \theta_j} \tag{2}$$

Thus, we define a probability distribution over possible configurations of observed and hidden variables, which enables us to explore the joint space of variable assignments in a probabilistic fashion.

3.2 Model Structure

Each document d is defined as a tuple $d = \langle \boldsymbol{w}, \boldsymbol{t}, \boldsymbol{m}, \boldsymbol{c} \rangle$ comprising an observed sequence of tokens $\boldsymbol{w}$ together with hidden sequences of non-overlapping entity mentions $\boldsymbol{m}$ and corresponding concepts $\boldsymbol{c}$. Further, we capture possible semantic transformations $\boldsymbol{t}$ as hidden variables that are intended to capture (near-)synonymy of individual tokens. Semantic transformations can be applied to observed input words in order to facilitate the normalization step in cases where a surface mention and a concept name differ by one synonymous token (e.g., “kidney dysfunction” vs. “kidney disease”). Each annotation span can have

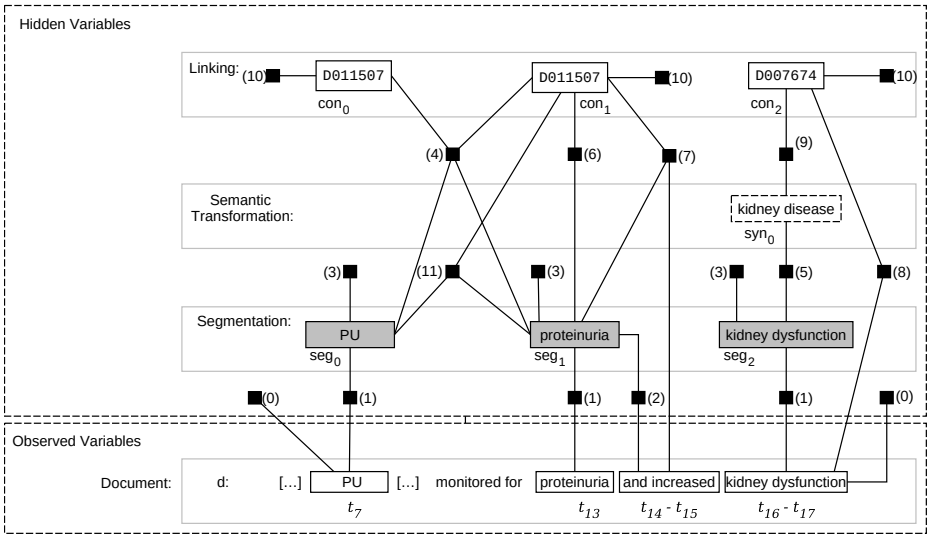

Fig. 1. Simplified factor graph for a correctly annotated document. The figure shows all different types of factors (small black boxes) that are used in order to link observed and hidden variables. Hidden variables comprise concept variables (nodes labeled as con_i), semantic transformation variables (syn_i), and segmentation (recognition) variables (seg_i). Individual factor types are numbered to be referenceable (cf. Section 3.6). The approach tackles both tasks, recognition and linking, thus the only observed variables are the tokens from the pre-tokenized document content marked as t_i.

only one semantic transformation and must have at least one token that was not semantically transformed. Fig. 1 shows a (simplified) factor graph representation of our model for an example document.

We define one specific assignment of values to these variables in a document as a *state*. By applying Equation (2), we can compute the probability of each state, which will be exploited during inference and learning.

3.3 Inference

In order to assign values to the hidden values in the model, i.e., recognize token spans corresponding to entity types of interest and link them to knowledge base identifiers, we perform approximate inference following a Markov Chain Monte Carlo (MCMC) sampling scheme [21]. In MCMC sampling, the goal is to construct an approximation that is maximally close to the posterior distribution of interest, while sharing the factorization properties as defined by the factor graph [8]. This is achieved by generating a sequence of *states*, each of which corresponds to an assignment of a value to all (or a subset of) the variables in the model (cf. Section 3.2). Thus, by performing a local search, this procedure successively explores the search space of variable assignments for a given document.

Exploring the Search Space. Initially, an empty state s_0 is generated for each document, which can be modified in subsequent sampling steps. In each iteration, an annotation span explorer and a concept assignment explorer are consecutively applied in order to generate a set of proposal states which differ from the current state in one atomic change. The annotation span explorer is able to add a new non-overlapping (empty) annotation[4], remove an existing annotation, or apply a semantic transformation to one token. The concept assignment explorer can assign a concept to an empty annotation, or change or remove one from an non-empty annotation.

Applying these explorers in an alternating consecutive manner effectively guarantees that all variable assignments are mutually guided by several sources of information: (i) Possible concept assignments can inform the annotation span explorer in proposing valid spans over observed input tokens, while (ii) proposing different annotation spans together with semantic transformations on these may facilitate concept linking. Thus, this intertwined sampling strategy effectively enables joint inference on the recognition and the linking task. In order to illustrate the sampling procedure, Fig. 2 shows a subset of proposal states as generated by the annotation span explorer.

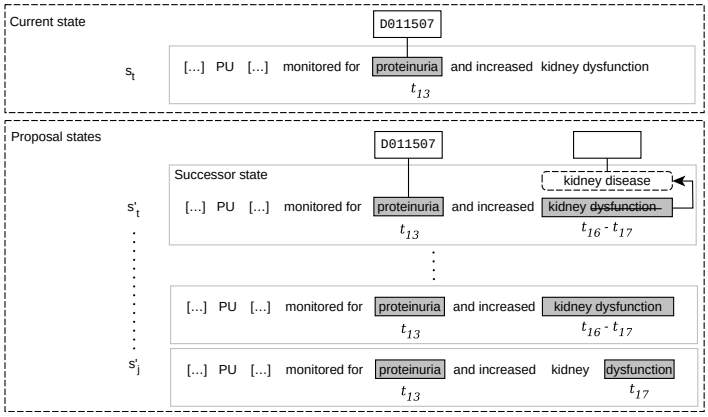

Fig. 2. Subset of proposal states generated by the annotation span explorer, originating from the current state s_t which has already one annotated span on token t_{13}. Each proposal state has a new non-overlapping segment annotation (marked in grey). Proposal states may include semantic transformations (depicted as dashed boxes). As shown for s'_t, new annotations have an empty concept assigned. Semantic transformations in a successor state are accepted for all subsequent sampling steps.

[4] We do not extend or shrink existing spans. Instead, new annotations can be of different length, spanning 1 to 10 tokens.

Evaluating States. From the set of all generated proposal states, we select one state s_{t+1} to be used as the successor state in the subsequent sampling step, following Hakimov et al. [5]. States are evaluated according to their individual probability (cf. Equation 2). But, the possible successor state s'_t is only accepted if its probability is higher than the probability of the current state s_t[5]:

$$s_{t+1} = \begin{cases} s'_t, & \text{if } p(s'_t) \geq p(s_t) \\ s_t, & \text{otherwise} \end{cases} \tag{3}$$

3.4 Parameter Learning

The learning problem consists in finding the optimal weight matrix θ that maximizes the probability of a sequence of assigned entity labels given observed training sequences (cf. Equation 2). We use SampleRank [24] to learn these parameters based on gradient descent on pairs of states (s_t, s'_t) that are investigated in individual steps of the inference procedure. Two states are compared according to the following preference function $\mathbb{P} : S \times S \rightarrow \{0, 1\}$:

$$\mathbb{P}(s', s) = \begin{cases} 1, & \text{if } \mathbb{O}(s') > \mathbb{O}(s) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Here, $\mathbb{O}(s)$ denotes an objective function that returns a score for s indicating its degree of accordance with the ground truth annotations in the respective training document in terms of the proportion of correctly linked entities and the total number of gold entity mentions in s (cf. [5]).

3.5 Dictionary Generation and Candidate Retrieval

Dictionary Generation A main component of our approach is a dictionary $\delta \subseteq C \times S$, where $C = \{c_0, \ldots, c_n\}$ is the set of concepts from a reference knowledge base and $S = \{s_0, \ldots, s_m\}$ denotes the set of names that can be used to refer to these concepts. We define two functions on the dictionary: (i) $\delta(s) = \{c \mid (c, s) \in \delta\}$ returns a set of concepts for a given name s, and (ii) $\delta(c) = \{s \mid (c, s) \in \delta\}$ returns a set of names for a given concept c.

Synonym Extraction. We extract a synonym lexicon from the dictionary δ by considering all names of a concept c that differ in one token. We consider these tokens as synonyms. For example, the names *kidney disease* and *kidney dysfunction* are names for the same concept and differ in the tokens 'disease' and 'dysfunction'. The pair (*disease*, *dysfunction*) is then inserted into a synonym lexicon denoted as σ provided that the pair occurs in at least two concepts.

Concept Candidate Retrieval. Candidate retrieval identifies, for each annotated segment, a number of concept candidates that the segment can denote. We implement the candidate retrieval using an index for the dictionary δ that maps names to concepts. The index is implemented using Lucene[6]; results are ranked

[5] We stop the inference procedure if the state does not change for 3 times in a row.
[6] https://lucene.apache.org/

using the built-in Lucene similarity score. We retrieve the top k candidates with a similarity of at least λ.

3.6 Templates and Feature Generation

As shown in Fig. 1, our model is designed by 12 individual types of factors (henceforth numbered between 0 and 11). We distinguish factors by their scope, i.e., whether they are used for the recognition or the linking task or jointly for both. Recognition factors are either connected to a single observed variable (type 0), or connect two variables of type *segmentation* or *synonym* (1, 2, 3 and 5). All these factors contribute features for the recognition task. Being connected to a single hidden variable of type *concept*, factor (10) has a scope that is limited to the linking task. Joint factors (4, 6, 7, 8 and 11) connect at least one variable of type *concept* with at least one variable of a different type.

Although factors can be grouped by their scope, we decide to apply a more semantic grouping of factors in our implementation. In the following, we describe our design of templates capturing the semantic relatedness of factors. All described features are of boolean type. Henceforth, we use d to denote the current document, s_i to denote the ith annotation span in d and c_i to denote the concept assigned to s_i. Further, the templates make use of the dictionary δ, and the semantic lexicon σ as previously described in Section 3.5. For readability, we introduce the abbreviations *seg*, *sem* and *con* for the three types of hidden variables: segmentation, semantic transformation, and concept, respectively.

Dictionary Lookup This template adds factors of type 1, 5, 6 and 9 to the factor graph. A feature of this template indicates whether s_i corresponds to an entry in the dictionary δ, i.e., whether $(s_i, c') \in \delta$ for some c'. A further set of features specific for each concept c_i indicates whether the span of an annotated entity mention refers to concept c_i according to the dictionary, i.e. $(s_i, c_i) \in \delta$. A further set of features indicate whether the semantically transformed version of s_i is in the dictionary or denotes concept c_i according to the dictionary.

Semantic Transformation The semantic transformation template adds a new factor of type 5 connecting a variable of type *sem* with a *seg* variable. The feature indicates for a given synonym pair (t_j, t'_j) whether *seg* corresponds to the semantically transformed version *sem* modulo the fact that some token t_j in *seg* is replaced by t'_j in *sem*.

Token Length This template connects a factor of type 3 to a *seg* variable. The factor defines n_i features indicating whether the number of tokens in *seg* is lower or equal than n_i where n_i is the number of tokens in s_i.

Token Context This template extends the factor graph by factors 2 and 7. It introduces three types of context features indicating if a span (i) is preceded

by a certain n-gram, (ii) is followed by a certain n-gram, and (iii) whether it is preceded and followed by a pair of n-grams ($1 \leq n \leq 4$). In addition, each of these features is conjoined with a specific concept c_i that the span is linked to.

Annotation Prior This template extends the factor graph by factor types 0, 8 and 10. The features provide a context-independent prior derived from training data indicating whether a segment s_i appearing in the training data represents a mention of an entity. Another set of features are concept-specific and indicate whether a segment s_i appearing in the training data represents a mention of an entity denoting concept c_i. In addition to considering the whole segment, we also consider n-grams ($1 \leq n \leq |s_i| - 1$).

Coherence This template adds a factor of type 4 which measures the coherence of annotations. Given all *seg* variables with the same mention text, we record whether all these variables are annotated with the same concept.

Abbreviation In this template, we address the problem of abbreviations (cf. [6]) in the task of entity linking. The template adds three types of factors 3, 6 and 11, where each factor has exactly one feature. Factor 3 is connected to a segmentation variable. The corresponding feature indicates whether the mention text represents an abbreviation[7] that occurs in the training data. Factor 6 connects segmentation variables with concept variables. Its feature indicates whether the given mention text is an abbreviation for the given concept according to the training data. Factor 11 connects two or more segmentation variables with a concept variable. Its feature measures whether there is a longform annotated that has the same concept assigned as the annotation s_i.

4 Experiments

We state our problem as joint sequence labeling and resolution comprising named entity recognition and linking. The objective is to recognize segments in text denoting an entity of a specific type and linking them to a reference knowledge base by assigning a unique concept identifier. In this section, we describe our experiments on two types of biomedical entities. The first experiment evaluates our system in disease recognition and linking. The second experiment is conducted on chemicals. Both experiments use the same data set described below.

4.1 Data Sets and Resources

Data Sets. All experiments were conducted on data from the BioCreative V Shared Task for Chemical Disease Relations (BC5CDR) [23]. The data set was

[7] We define an abbreviation as a single token which is solely in uppercase and has at most 5 characters.

designed to solve the tasks of entity recognition and linking for disease and chemicals and further to find relations between both. However, the latter task is not yet considered in our approach. Each annotation contains information about its span in terms of character offsets and a unique concept identifier. Annotated entities are linked to the Comparative Taxicogenomics Database[8] for diseases (CTD_{dis}) or chemicals (CTD_{chem}), respectively. The data set consists of 1,500 annotated Pubmed abstracts equally distributed into training, development and test set with about 4,300 unique annotations each.

Reference Knowledge Base. CTD_{dis} is derived from the disease branch of MeSH and the Online Mendelian Inheritance in Man (OMIM)[9] data base. CTD_{dis} contains 11,864 unique disease concept identifiers and 75,883 disease names. CTD_{chem} is solely derived from the chemical branch of MeSH. It comprises 163,362 unique chemical concept identifiers and 366,000 chemical names.

Cleaning Procedure. In order to remove simple spelling variations, we implement a text cleaning procedure which is applied to all textual resources and data sets. The strategy uses six manually created regular expressions like replacing *'s* by *s*. Further, we convert all tokens into lowercase if they are not solely in uppercase, we remove all special characters including punctuation and brackets, and replace multiple whitespace characters by a single blank. We apply the same strategy to both diseases and chemicals.

Resources used in the Experiments. In the experiments for disease recognition and linking, we initialize the dictionary δ with CTD_{dis} and enhance it with the disease annotations from the training data. We then apply the text cleaning procedure as described above to all entries, as well as to all documents in training and test set. Due to the cleaning, the size of the dictionary reduces to 73,773 unique names ($-$2,113), while the number of concepts remains the same. The resulting synonym lexicon σ stores 2,366 entries.

In the experiments for chemicals, the dictionary δ is initialized with CTD_{chem} and enhanced with the chemical annotations from the training data. After the cleaning procedure, the size of the dictionary reduces to 359,564 unique names ($-$8.186), while the number of concepts remains the same. The resulting synonym lexicon σ stores 4,912 entries.

The system's overall performance depends on the two parameters k and λ that influence the candidate retrieval procedure (cf. Section 3.5), as they determine the maximum recall that can be achieved. We empirically set the best parameter values using a two-dimensional grid search on the development set, assuming perfect entity recognition. Best performance is achieved with $k = 20$ and $\lambda = 0.7$. Given these parameters, a maximum recall of 90.4 for diseases, and 91.5 for chemicals can be obtained by our system on the BC5CDR test set.

[8] http://ctdbase.org, version from 2016.
[9] http://www.omim.org

4.2 Baselines

We compare our approach to the two state-of-the-art systems *DNorm* [12] and *TaggerOne* [11], as well as against two simple baselines (*LMB* and *LMB*$^+$). The latter baselines are based on non-overlapping longest matches, using the dictionary as described in Section 3.5. While in *LMB*$^+$ all resources (including the dictionary and documents) were cleaned, resources in *LMB* remain as they are. Due to the cleaning, we lose track of the real character offset position. Thus, these baselines are not applicable to the entity recognition subtask.

4.3 Experimental Settings

Evaluation Metrics. We use the official evaluation script as provided by the BioCreative V Shared Task organizers [23]. The script uses Precision, Recall and F_1 score on micro level. In the recognition task the measure is on mention level comparing annotation spans including character positions and the annotated text. Experiments on the linking task are evaluated on concept level by comparing *sets* of concepts as predicted by the system and annotated in the gold standard, i.e., multiple occurrences of the same concept and their exact positions in the text are disregarded.

Hyper-Parameter Settings. During development, the learning rate α and the number of training epochs ϵ as hyper-parameters of SampleRank were empirically optimized by varying them on the development set. Best results could be achieved with $\alpha = 0.06$. The results reached a stable convergence at $\epsilon = 130$.

4.4 Results

We report results on the BC5CDR test set in Table 1. Results on the disease and chemicals subtasks are shown in the left and right part of the table, respectively. For both tasks, we assess the performance of our system on end-to-end entity linking (columns labeled with "Linking"), as well as the entity recognition problem in isolation ("Recognition").

Disease Recognition and Linking. In disease recognition, our approach exhibits the best F_1 score of all systems compared here (F_1=83.2). Only in terms of Precision, TaggerOne has slight advantages.

In the linking task, our system clearly outperforms both lexicon-based baselines as well as both state-of-the-art systems. In particular, J-Link exceeds TaggerOne by 2.2 and DNorm by 5.3 points in F_1 score, respectively.

Comparing these results to the baselines, we observe that a simple lexicon lookup (LMB) already achieves robust precision levels that cannot be met by the DNorm system. More than 22 points in recall can be gained by simply applying a cleaning step to the dictionary and documents (LMB$^+$). However, the increasing recall comes with a drop in precision of 1.8 points. This shows that preprocessing the investigated data can be helpful to find more diseases, while aggravating the

Table 1. Evaluation results on BC5CDR test set for recognition and linking on diseases (left part) and chemicals (right part)

	Diseases						**Chemicals**					
	Recognition			Linking			Recognition			Linking		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
J-Link	84.6	**81.9**	**83.2**	**86.3**	**85.5**	**85.9**	90.0	86.6	88.3	85.9	**91.0**	88.4
TaggerOne	**85.2**	80.2	82.6	84.6	82.7	83.7	**94.2**	**88.8**	**91.4**	88.8	90.3	**89.5**
DNorm	82.0	79.5	80.7	81.2	80.1	80.6	93.2	84.0	88.4	**95.0**	80.8	87.3
LMB^+	n/a	n/a	n/a	80.5	80.9	80.7	n/a	n/a	n/a	80.4	82.7	81.5
LMB	n/a	n/a	n/a	82.3	58.5	68.3	n/a	n/a	n/a	84.0	58.8	69.2

linking task. Obviously, our system (in contrast to DNorm and to a greater extent than TaggerOne) benefits from a number of features that provide strong generalization capacities beyond mere lexicon matching. A more detailed feature analysis is deferred until Section 4.5.

Chemicals Recognition and Linking. In the second experiment, we are interested in assessing the domain adaptivity of our model. Therefore, we apply the same factor model to a different reference knowledge base, without changing any system parameters or engineering any additional domain-specific features.

The evaluation (cf. Table 1, right part) shows promising results regarding the adaptation to chemicals, particularly in the linking task. Our approach is competitive to DNorm and TaggerOne, while clearly outperforming both lexicon baselines. Compared to DNorm, our approach lacks in precision (−9.1), but shows better results in recall (+10.2), which results in a slightly higher F_1 score (+1.1). Overall, TaggerOne obtains the best performance in this experiment, due to the best precision/recall trade-off. However, the superior recall of our system is remarkable (R=91.0), given that the dictionary for chemicals as used in TaggerOne was augmented in order to ensure that all chemical element names and symbols are included [11].

4.5 Discussion

Comparison to Previous Work To our knowledge, the best performance in previous work on disease linking has been obtained by Lee et al. [13] who report an F_1 score of 86.5 (P=89.6; R=83.5) on the BC5CDR test set. While these results are slightly higher than the ones we report in Table 1, their system benefits from two design choices that are highly task-specific: First, the authors extend their lexicon annotations from the NCBI Disease corpus [1]. Further, they manually extend the dictionary underlying their system to account for synonym variations in the corpus. We apply automatically learned semantic transformations to this problem. Third, their dictionary lookup follows a fixed sequential order in which the lexical resources are consulted. This order is optimized to the disease linking task on BC5CDR data. In contrast, we aim at a general model

for joint entity recognition and linking that can be flexibly applied to existing knowledge bases of various domains, without the need of manual adaptations.

Upper Bounds The upper bound of our approach is determined by the maximum recall of the candidate retrieval. Given the optimized parameters $k = 20$ and $\lambda = 0.7$ (cf. Section 4.3), our upper bound is limited to R=90.4 for disease linking. Compared to our observed recall performance in the linking task, there are 4.9 points left for improvement. Keeping k and λ for the chemical linking task, we reach the upper bound in recall with a delta of only 0.5 points. Thus, a further increase in recall can only be obtained by varying the candidate retrieval at the cost of generating a larger amount of spurious candidates.

Template Ablation We investigated the impact of the individual templates in an ablation test. The resulting Δ in F_1 (in comparison to the full model) is shown in Table 2. All evaluations were done on the development set for diseases, using the previously described settings.

Table 2. Impact of individual templates to overall performance in disease recognition and linking, according to an ablation test on the development set. Results are separated into recognition (left) and linking (right).

	Recognition				Linking			
Configuration	Prec.	Recall	F_1	ΔF_1	Prec.	Recall	F_1	ΔF_1
All Templates	83.9	76.9	80.2		85.6	80.7	83.1	
−Annotation Prior	81.5	77.0	79.2	−1.0	81.6	80.8	81.2	−1.9
−Abbreviation	83.3	76.8	79.9	−0.3	85.3	80.9	83.0	−0.1
−Coherence	84.2	76.3	80.0	−0.2	85.1	80.4	82.6	−0.5
−Token Context	84.2	74.4	79.0	−1.2	86.3	79.2	82.6	−0.5
−Token Length	81.3	72.7	76.8	−3.4	83.9	76.5	80.0	−3.1
−Lexicon	76.2	46.6	57.9	−22.3	87.2	55.1	67.6	−15.5
−Sem. Transform.	84.0	75.4	79.5	−0.7	85.7	79.1	82.3	−0.8

As can be seen from the table, the relative impact of templates on recognition and linking follows a largely consistent pattern. As for disease recognition, the strongest increase in F_1 is due to the *Lexicon* template ($\Delta F_1 = -22.3$). In disease linking, this template heavily increases recall but leads to slight drop in precision ($\Delta F_1 = -15.5$). The *Token Length* template equally increases recall and precision in both tasks. Its impact is the second highest, which can be explained by its broad scope. Although the *Annotation Prior* has a similarly broad scope, its impact is smaller as we added the training data to the dictionary, which leads to a partial subsumption of this template by the *Lexicon* template. *Abbreviation*, *Coherence* and *Semantic Transformation* templates have a rather restricted scope in that they address very specific phenomena. Our evaluation shows that adding

these templates does not negatively interfere with other templates, but increases either recall, precision, or both.

Error Analysis Typical errors of our system are due to incorrectly resolved abbreviations, erroneous span detection during recognition (e.g., *infection by hepatitis B virus* vs. *infection*), fine-grained semantic distinctions during linking (e.g., (i) terms such as *seizures* or *shock* which exactly match an entry in the dictionary, but are not annotated as diseases in the data, or (ii) distinctions between *psychological* or *physiological* diseases, or *substance-induced*, *acute*, or *chronic* diseases), and discrepancies in the annotated training and testing data.

5 Conclusion

We have presented a domain-independent system that jointly addresses both the entity recognition and linking task using a probabilistic framework. The framework builds on an undirected probabilistic graphical model that considers the span of mentions of entities as well as the corresponding knowledge base identifier as random variables and models the joint assignment using a factorized distribution. We have shown that our approach can be easily applied to different domains by merely exchanging the underlying ontology and training data. On the task of recognizing and linking disease names, we show that our approach outperforms the state-of-the-art systems *DNorm* [12] and *TaggerOne* [11], as well as two lexicon-based baselines. On the task of recognizing and linking chemical names, our system achieves comparable performance to the state-of-the-art.

In future work, we plan to corroborate the domain adaptivity of our system by investigating different entity types beyond diseases and chemicals. Moreover, applying J-Link to *simultaneously* linking entities of *multiple* types (as demonstrated by [11] for the two types of diseases and chemicals) would be a promising avenue towards semantic representation of large heterogeneous text collections.

Acknowledgments

This work has been funded by the Federal Ministry of Education and Research (BMBF, Germany) in the PSINK project (project number 031L0028A).

References

1. Doğan, R.I., Leaman, R., Lu, Z.: NCBI Disease Corpus. A Resource for Disease Name Recognition and Concept Normalization. Journal of Biomedical Informatics 47, 1–10 (2014)
2. Durrett, G., Klein, D.: A Joint Model for Entity Analysis. Coreference, Typing, and Linking. TACL 2, 477–490 (2014)
3. Finkel, J.R., Grenager, T., Manning, C.: Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. In: Proceedings of ACL. pp. 363–370 (2005)

4. Ganea, O.E., Ganea, M., Lucchi, A., Eickhoff, C., Hofmann, T.: Probabilistic Bag-Of-Hyperlinks Model for Entity Linking. In: Proc. of WWW. pp. 927–938 (2016)
5. Hakimov, S., ter Horst, H., Jebbara, S., Hartung, M., Cimiano, P.: Combining Textual and Graph-based Features for Named Entity Disambiguation Using Undirected Probabilistic Graphical Models. Knowledge Engineering and Knowledge Management pp. 288–302 (2016)
6. Hartung, M., Klinger, R., Zwick, M., Cimiano, P.: Towards Gene Recognition from Rare and Ambiguous Abbreviations using a Filtering Approach. In: Proceedings of BioNLP 2014. pp. 118–127 (2014)
7. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust Disambiguation of Named Entities in Text. In: Proceedings of EMNLP. pp. 782–792 (2011)
8. Koller, D., Friedman, N.: Probabilistic Graphical Models. Principles and Techniques. MIT Press (2009)
9. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor Graphs and Sum Product Algorithm. IEEE Transactions on Information Theory 47(2), 498–519 (2001)
10. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields. Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of ICML. pp. 282–289 (2001)
11. Leaman, R., Lu, Z.: TaggerOne. Joint Named Entity Recognition and Normalization with Semi-Markov Models. Bioinformatics 32, 2839–46 (2016)
12. Leaman, R., Dogan, R.I., Lu, Z.: DNorm. Disease Name Normalization with Pairwise Learning to Rank. Bioinformatics 29, 2909–2917 (2013)
13. Lee, H.C., Hsu, Y.Y., Kao, H.Y.: An Enhanced CRF-Based System for Disease Name Entity Recognition and Normalization on BioCreative V DNER Task. In: Proceedings of the BioCreative V Workshop. pp. 226–233 (2015)
14. Luo, G., Huang, X., Lin, C.Y., Nie, Z.: Joint Entity Recognition and Disambiguation. In: Proceedings of EMNLP. pp. 879–888 (2015)
15. Mihalcea, R., Csomai, A.: Wikify! Linking Documents to Encyclopedic Knowledge. In: Proc. of CIKM. pp. 233–242 (2007)
16. Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation. A Unified Approach. TACL 2, 231–244 (2014)
17. Nadeau, D., Sekine, S.: A Survey of Named Entity Recognition and Classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
18. Nguyen, D., Theobald, M., Weikum, G.: J-NERD. Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features. TACL 4, 215–229 (2016)
19. Poon, H., Domingos, P.: Machine Reading: A "Killer App" for Statistical Relational AI. In: Proc. of StarAI. pp. 76–81 (2010)
20. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of ACL:HLT. pp. 1375–1384 (2011)
21. Singh, S., Wick, M., McCallum, A.: Monte Carlo MCMC. Efficient Inference by Approximate Sampling. In: Proceedings of EMNLP. pp. 1104–1113 (2012)
22. Usbeck, R., Ngomo, A.C.N., Röder, M., Gerber, D., Coelho, S.A., Auer, S., Both, A.: AGDISTIS. Graph-based Disambiguation of Named Entities Using Linked Data. The Semantic Web–ISWC 2014 pp. 457–471 (2014)
23. Wei, C.H., Peng, Y., Leaman, R., Davis, A.P., Mattingly, C.J., Li, J., Wiegers, T.C., Lu, Z.: Overview of the BioCreative V Chemical Disease Relation (CDR) Task. In: Proc. of the BioCreative V Evaluation Workshop. pp. 154–166 (2015)
24. Wick, M., Rohanimanesh, K., Culotta, A., McCallum, A.: SampleRank. Learning Preferences from Atomic Gradients. In: Proc. of the NIPS Workshop on Advances in Ranking. pp. 1–5 (2009)