Document Annotation Support Using Biomedical Ontologies

Tran Thanh[1], Yongtao Ma[1], Natalya F. Noy[2]

[1] Institute AIFB, Karlsruhe Institute of Technology, Germany, tran,ma@kit.edu
[2] Stanford Center for Biomedical Informatics Research, Stanford University, US, noy@stanford.edu

Abstract. Adding annotations to documents by extracting data from text yields richer document representation, which users can exploit for various tasks such as search and browsing. However, data extraction is hard, especially in large-scale heterogeneous settings. A more focused technique for data extraction is *entity linking*, which does not extract new data from documents, but links words in documents to referent entities in existing structured datasets or ontologies. We follow this direction in the sense that given some documents, we also aim at finding entities (ontology concepts in our case) that can be used for creating document annotations. However, we emphasize the role of users in this annotation creation process such that the concepts we search for are not directly annotations, but candidate annotations as well as their contexts forming annotation modules that are then employed by users for creating the annotations manually. We propose a technique, which efficiently computes annotation candidates based on a coarse-grained topic-based representation of documents and ontology concepts. Aiming at maximizing compactness while preserving useful information, we also elaborate on a module extraction technique, which considers only annotation candidates and context elements that are "on-topic", i.e. share topics with the documents to be annotated. Initial experiments show promising results as well as the needs for much more research work towards this new direction of ontology-based annotation support.

1 Introduction

Researchers have extensively studied the task of extracting data from text given a large collection of textual documents. The goal of this task is to obtain a deeper understanding of the text and to represent it in terms of structured data. Enriching the traditional bag-of-words representation of text with structured data in the form of *annotations*, can be beneficial for various document-management tasks. For instance, a conceptual representation of text has been used for search and retrieval to bridge the semantic gap between the text and user information needs [17, 10]. Annotations can also help in browsing as well as understanding documents [9]. One specific domain where document annotations are extensively used is biomedicine. One part of the NCBO BioPortal [3] project for instance, is

[3] http://bioportal.bioontology.org/

dedicated to annotating large collections of documents covering different aspects in the biomedical domain (e.g. PubMed, ClinicalTrials.gov and DrugBank) with concepts captured by biomedical ontologies (e.g. SNOMED and NCI Thesaurus).

Researchers have explored different directions for extracting data from text to create annotations. High quality results can be obtained by using manually specified rules or sophisticated machine learning techniques for inducing extraction patterns, and by combining different extraction programs [3]. Yet, scaling these solutions to the large-scale setting where the amount of text, data and data extraction patterns is large, and where training data are difficult to obtain, has proven to be a hard problem. Thus, recently, researchers have started incorporating input and feedback from human experts into the extraction process [2]. While also following this direction, our work presented here puts the human experts into the center of the machinery. That is, the main driving force is the human experts, who are supported by computer programs, but not the other way around. Our work is aimed at facilitating the participation of human experts in this process, by automatically extracting the relevant concepts and contextual information from ontologies, which can be used by experts for creating the annotation manually, thus reducing the cognitive complexity of their task: they will need to look at a much smaller set of possible concepts.

In particular, we are not dealing with the general data extraction problem but specifically focus on the problem of creating annotations using available ontologies. Our work is motivated by the fact that the amount of ontologies available is continuously increasing. Trends in publishing and making them publicly available on Web suggest that most concepts that are captured by textual documents are already available in ontologies. Instead of extracting new concepts from text, we can partially reduce the problem of bridging the gap between the textual and the conceptual representation to the more tractable problem of linking text elements to ontology concepts (or vice versa). In this work, we aim at supporting human experts in creating these links (i.e. to create document annotations) using available ontologies. Given a collection of documents, we want to find parts of the ontologies (called modules), which contain concepts that can be used for annotating these documents. Ideally, the computed module(s) contain these candidate concepts as well as contextual elements that can help users in understanding and linking these concepts to the corresponding part in the text.

In summary, the contributions of this paper are as follows: (1) We explore the new avenue of *supporting the manual annotation creation process by extracting relevant modules from available ontologies*. To the best of our knowledge, this is a novel problem that has not been studied before. The technical challenge behind it is scale because the amount of ontologies is large and as a consequence, enumerating, extracting and analyzing all possible candidate modules are computationally expensive tasks. (2) We propose to use a *coarse-grained representation of documents and ontology concepts* based on topics. Based on the well-studied generative model called Latent Dirichlet Allocation (LDA) [1], we learned topics from textual documents and "textual representation" of ontology concepts. Then, given the set documents to be annotated and its topical repre-

sentation, we identify concepts that are about the same topics. These candidate annotations are then used for constructing modules, which additionally, also contain contextual information. (3) We develop an *algorithm that searches for cliques in the ontology graph that contain both the candidate annotations and the paths connecting these annotations*. (4) We report on *preliminary experiments*, which promisingly show that the coarse-grained topic-level representation helps to improve efficiency while not comprising too much on quality when compared with a more sophisticated baseline operating at the word-level. Also, they show that the proposed algorithm yields "on-topic" modules, that is, modules that contain contextual information that are of the same topics as the documents to be annotated.

2 Related Work

Researchers have extensively studied the problem of extracting concepts, named entities, and complex relations between entities from text. We can roughly classify existing approaches into two categories: (1) there are declarative rule-based approaches [5, 6, 11] such as the one implemented by IBM's SystemT [4], and there are learning-based approaches employing classifiers (e.g. based on conditional random fields [13]).

The problem that we are investigating is not general data extraction. Rather, our problem is similar to the more specific problem studied under the notion of entity linking. Given entities captured by a structured dataset, the goal here is to find corresponding words in the text, i.e. linking entities in data with their referents in the text (or vice versa) [15, 7, 8]. Recently, a generative probabilistic model has been proposed to deal with this task. Exploiting three different information sources, namely popularity of entities, entity names, and entity contexts, the presented solution learns three distributions, and then combine evidences derived from them to find the referent entity in the data of a name mention in the text [7]. Also, collective entity linking has been proposed [7], which leverages the fact that entities in the same document should be semantically related to each other. Thus, interdependence between name mentions are exploited to jointly link them with their referent entities. The difference to this line of work is that we emphasize the role of human experts. The linking between ontology concepts and text elements is not performed automatically but manually by experts, and the goal of this work is to support them by creating ontology modules containing all (and only) the elements that are relevant for this task. This direction of incorporating human experts into the loop [2] aims at high-quality results, which are hard to obtain in complex, heterogeneous and large-scale scenarios. In order to realize this kind of annotation support, the underlying problem to be solved is in principle similar to entity linking because firstly, annotation candidates have to be found in the ontologies. However, the inputs for this task are not a number of name mentions in the text (or concepts in the ontologies) but a collection of documents that shall be annotated. Because all words in the documents and all concepts in the ontologies are possible candidates for pairwise linking, the

search space is large, thus calling for a more scalable solution. The approach we elaborate on in this paper does not operate at the the level of words and concepts, but at the more coarse-grained level of topics. Also, the outputs do not comprise links between name mentions in the documents and specific concepts in the ontologies but rather, links between documents and modules that are part of the ontologies.

Furthermore, the problem of annotation support goes beyond the entity linking problem because the resulting modules should contain not only annotation candidates but also additional elements that can help experts in understanding these candidates and their contexts. Thus, the second subproblem to be solved is related to ontology modularization. Researchers have studied a large number of modularization approaches, ranging from approaches that look at the computational properties of the modules, support interactive techniques, or use automatic module extraction [16]. To the best of our knowledge, our approach is the first one that uses the associated text and the topical content of ontologies to extract modules that are about certain topics.

3 Overview

We focus our work in the biomedical domain. There is a large number of document collections we want to annotate using knowledge captured by the vast amount of existing biomedical ontologies that are made available through the NCBO BioPortal. These ontologies vary, ranging from simple taxonomies of concepts to complex logic-based models. For reasons of generality and simplicity, we employ a generic graph-based model that omits the specific (formal) semantics captured by some of the available ontologies.

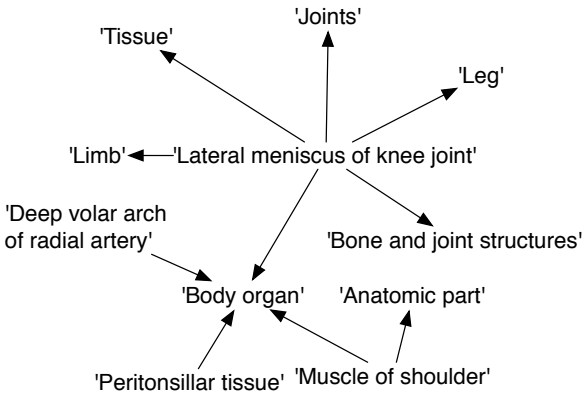


Fig. 1. Extract of an Ontology

Data. We view each ontology as a directed and node-labeled graph $G = (N, E)$ (Fig. 1). Nodes N stand for concepts and edges E capture different types of semantic relationships between concepts. In fact, the specific semantics behind the nodes and edges are not relevant in our work as we exploit only node *labels* and *structure* information. An extract of an ontology where nodes are labeled, and connected by some edges, is shown in Fig. 1. A *module* M of G is simply a subgraph of G. Besides ontologies and modules, we have collections of *documents* D_i where every document $d \in D_i$ is conceived as a bag of words, i.e., $d = \{t_1, \ldots, t_{|d|}\}$.

Problem. Our goal is to support domain experts in creating *annotations*. An annotation is a tuple, which associates an ontology graph node with a word (a sequence of words) in the document. Here, we are not interested in computing the annotations but it suffices to determine *annotation candidates*—ontology-graph nodes that can be associated with a document and used by the expert to annotate it. Further, we aim to extract modules from the ontology. Intuitively, an *optimal module* shall (1) maximize the number of ontology-graph nodes that can potentially be used for creating annotations (i.e., relevant nodes), and (2) minimize the amount of superfluous information. Accordingly, the most straightforward strategy is to compute and to return all and only those ontology nodes that can be linked to document words. The resulting module comprises relevant nodes only. However, while such a modularization strategy can minimize the amount of superfluous information, it does not take into account the information that is necessary for users to understand these relevant candidates. It does not capture the context where they come from. Thus, an optimal module contains all relevant annotation candidates as well as their context elements. Clearly, which contextual elements are useful and which ones are superfluous eventually can only be decided by the actual user. This aspect of usefulness is crucial and will be investigated in future work. In this initial work, we focus on the technical aspect and address the following problem: Given the ontology G, the collection D, and a subset of documents $d \in D$ to be annotated, we are interested in finding modules M in G, which contain all annotation candidates and their contexts.

Solution. Our idea of computing optimal modules in that sense rests on the notion of topics, which stand for groups of words that appear in similar contexts. We learn topics for both collection documents and ontology concepts. Once the topic-based representation is available, we can classify concepts and documents into different topics. These topics are seen as coarse-grained features of documents and concepts. Therefore, when we compute annotation candidates, we can focus only on ontology nodes and documents that share the same topics. After computing the candidates, we explore paths between them in the ontology graph to construct modules. In summary, we can decompose this process into three main steps: (1) constructing a topic-based representation of documents and concepts, (2) deriving annotation candidates from concepts and documents, which share the same topics, and, finally, (3) extracting modules from the ontologies.

4 Topic-based Representation of Documents and Concepts

Given the ontology G and the collection of documents D, computing annotation candidates in the straightforward way requires $|G| \times |D|$ similarity computations, where $|G|$ is the number of ontology nodes and $|D|$ is the number of documents. Then, candidates in G can be selected based on their computed similarities to documents in D. In fact, a more effective solution (e.g. when using entity linking approaches for this task) may require comparisons at the level of words instead of documents. Clearly, this computation is expensive given that ontologies can have hundreds of thousands of concepts (e.g., SNOMED CT) and many document collections contain millions of documents. Furthermore, ontology node labels are often short and ambiguous. This is similar to the classical information retrieval problem, where given short queries with few and ambiguous terms, and a large of number of candidate documents containing these terms, it is hard to identify which candidates are relevant. In our case, given documents to be annotated, we have to find out which ontology nodes are relevant (given their short labels).

4.1 Topic Models

To address these two problems, we propose the use of LDA-based topic models [1]. LDA is the state-of-the-art unsupervised technique for extracting topical information from large document collections. Given a set of unique words $\mathbf{V} = \{w_1, ..., w_{|V|}\}$, where V is the vocabulary, and each document in the corpus is a bag of words $w_i \in V$, a topic z_j, $1 \leq j \leq K$, is represented as a multinomial probability distribution over the words in V, $p(w_i \mid z_j)$, where $\sum_i^V p(w_i \mid z_j) = 1$. This distribution gives the probability a word in V will be generated from the topic, i.e., the probability it will be observed through the process of sampling repeatedly from V. Based on this notion, a document d is defined as the probability of generating $|d|$ words from a topic, which is analogous to picking a random word $|d|$ times from the vocabulary V. More precisely, the distribution over words given a document, $p(w_i \mid d)$, is actually defined as a mixture over topics, $p(z_j \mid d)$, associated with that document:

$$p(w_i \mid d) = \sum_{j=1}^{K} p(w_i \mid z_j) p(z_j \mid d) \tag{1}$$

where the parameters topic-word distribution, $p(w \mid z)$, and document-topic distribution, $p(z \mid d)$), are learned in a completely unsupervised manner (using statistical estimation techniques such as Gibbs sampling), without any prior knowledge of what words are associated with the topics or what topics are associated with the documents.

For our data in the biomedical domain, Table 1 shows topics, words associated with these topics, and the documents in which these words occur.

Researchers have shown that representing documents in terms of topics works well for keyword queries in traditional IR tasks. A topic in this sense is a group

Table 1. Example of Topics and Documents

Topic	Document
microbiologic, nitrates, cellular, urticaria, pretreatments, pistachios, mucous, intravascular, diluent, syringes, neurologic...	patients with MS are evaluated with a complete **neurological** examination and screening for heart disease
cell, skeletal, catabolism, peroxisomal, hypotonia, subpopulation, originates, gene...	The purpose of this study is to asses changes in **genetic** polymorphism's in **peroxisomal** proliferator... in glucose transporters 1 and 4 resulting in enhanced peripheral glucose utilization by fat and **skeletal** muscle.
lavage, intubation, reduction, aerosol, collaboration, trypsin, mic, stipulate, vortex, gloves, intubated, reaction, extensively...	that is either coughing or reacting to **intubation** attempts... E will consist of an N95 respirator, goggles, face shield, bouffant hair cover, fluid resistant surgical gown, and fitted sterile **gloves**.

of different words that occur in a similar context. Thus, a query and a document represented in such a lower-dimensional space can still have high similarity even if they do not share a word, or have low similarity when they share words that however, are used in different contexts (i.e. when their topics do not match). Note that the former case represents a vocabulary mismatch while the latter arises from the mismatch in semantics. Using topic-based representation yields performance that is superior than the state-of-the-art standard language modeling approach to IR [18], possibly due to its robustness in dealing with these types of mismatches.

We pursue this topic-based approach to deal with short and ambiguous concept labels. Also, it is used to address the scalability problem in our scenario, simply because similarity computation in this lower-dimensional space is less expensive.

4.2 Representing Data Using Topic Models

While generating topics from the actual documents is a straightforward application of LDA, it is not clear how to do so for the ontology nodes. Now, we elaborate on our solution to this problem.

Just like we did for the documents in a collection, we aim to represent ontology nodes as topic mixtures as defined in Equation 1. We propose the notion of *virtual document*. A virtual document is created from the labels of the ontology node and acts as its "textual proxy" based on which LDA is applied to generate topics.

In particular, recall that every node is associated with label information. In order to create a virtual document for a node, we firstly extract its labels

$k_1, \ldots, k_n$ to create a query $q = \{k_1, \ldots, k_n\}$. Given an information retrieval system, we retrieve all documents in the collection and obtain the top-k results for q. This local corpus analysis technique is similar to model-based feedback [20], where a retrieval run is performed against the corpus to obtain a set of feedback documents. Instead of using all the words in the feedback documents as "features" of the ontology node, a ranking on the words can be performed to select the top-k ones only. In this work, we rank features based on TF-IDF, i.e. count the word frequency TF in the feedback documents and normalize it using the inverse document frequency IDF derived from the entire collection. In the experiments performed in this work, we use 20 features per document on average, i.e., $k_{features} = 20$.

Given collection documents and virtual documents obtained for ontology nodes, we run LDA to obtain their topic-based representations. As usual, the number of topics is set as input to the LDA computation. For our experiments, we set it to 200, a common value, for which researchers have reported optimal performance in different settings.

5 Topic-based Computation of Candidate Annotations

As discussed before, the precomputed topics act as features, based on which annotation candidates are computed during this step. Given the collection of documents D and G during runtime, our goal here is to quickly filter out ontology nodes in G that are not relevant, and to focus on those nodes that can be used to annotate documents in D. We aim to do that by comparing ontology nodes and documents at the level of topics. This topic-based computation can be seen as a candidate filtering step similar to blocking [19], which has been proposed to deal with the scalability problem in entity consolidation. In particular, we classify virtual and collection documents to topics. Then, given the top-k topics Z_d obtained for every document $d \in D$, we retrieve ontology nodes $n \in G$ that share some topics with d, i.e. $Z_d \cap Z_n \neq \emptyset$.

Note that topics in the LDA model are hidden variables. Thus, the document-topic distribution, $p(z \mid d)$, as discussed previously is not explicitly available and therefore, cannot be used directly for assigning documents to topics. We now discuss how to classify documents to topics based on the topic-word distribution $p(w \mid z)$ returned from LDA instead.

Basically, the topic models $p(w \mid z)$ are multinomial distributions over words of the vocabulary. Given virtual and collection documents, we can classify them into topics, based on the probabilities document words are generated from these topic models. Intuitively speaking, a topic model assigns different levels of importance (probabilities) to words in the vocabulary. Based on that, we can classify a document d to a topic z, if d contains many words that are important for z. More formally, given Z is the set of all topics, $p(w \mid z)$ are topic models for $z \in Z$, we classify documents to topics simply based on the probability document words w in d are generated by topic z, i.e., $\sum_{w \in d} p(w \mid z)$. This way, we obtain a top-k set of topics for every document, sorted by the probability they generate document

words. In the evaluation, we vary k_{topics} to assess the sensitivity of the results w.r.t. this parameter.

6 Computing On-Topic Annotation Modules

The previous step produces a set of ontology nodes, which share some topics with the documents to be annotated. During this step, we address the problem of constructing modules that add contextual information to these annotation candidates. In general, the context of an ontology node n can be understood as its neighborhood, e.g. a subgraph of the ontology graph containing n, and all nodes and edges reachable from n via paths of length d or less. As discussed, which contextual information is actually useful may be personal and eventually, can only be determined by the individual users. We propose to exploit the topical information computed in the previous step and to derive the usefulness of contextual nodes based on the notion of topical relatedness. That is, ontology nodes provide useful contextual information w.r.t. annotation candidates when they are about the same topics.

In particular, we are interested in constructing modules, which contain as many annotation candidates as possible and additionally, satisfy the following properties:

On Topic: A path in the ontology graph can be conceived as a sequence of nodes N. Given the set of documents, D, and the set of topics associated with documents in D, Z_D, a path is called an on-topic path when the topics of its constituent nodes overlap with the topics of the documents, i.e. $\forall n \in N, Z_n \cap Z_D \neq \emptyset$. A module is on-topic if and only if all its constituents paths are on-topic.

Limited Length: A module M is limited in length when its longest path is of length d or less. That is, every node in M is reachable from every other node in M through a path of length d or less. This is a rather technical aspect of module construction because based on the parameter d, we can control the size of modules to be generated. It helps to avoid exploring the entire ontology graph (which could be very large in practice), given the desired size for a module is known in advance.

In order to construct such modules, we propose a bottom-up search strategy, which explores the ontology graph in both edge directions, i.e. G is treated as an undirected graph. Firstly, it (1) identifies *atomic on-topic modules*, (2) and then iteratively merges them until reaching the length limit.

An atomic on-topic module is simply a subgraph of the ontology graph, which comprises annotation candidates as computed before only. That is, nodes in that module make up a subset of the set of annotation candidates. The problem of extracting such modules can be formulated as the one of searching maximal cliques in the ontology graph G. A clique here is a subgraph of G, where each pair of nodes in the clique is directly connected by an edge. Such a clique is maximal when it cannot be extended by including any other adjacent nodes in the graph. Now, when we further set nodes in the clique to be only the ones

that are in the set of annotation candidates, then the resulting clique clearly satisfies the on-topic property. We adopt the Bron-Kerbosch's clique detection algorithm [12] to list all maximal cliques and to obtain atomic on-topic modules.

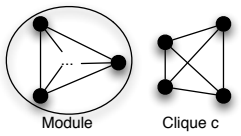


Fig. 2. Isolated Cliques

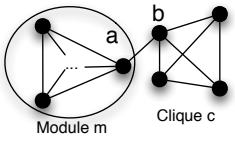


Fig. 3. Connected

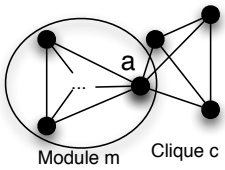


Fig. 4. Overlapping

Our technique for merging atomic modules is based on the following observation: there exist three different types of connections between a (combined or atomic) module M_d (left side of Figs. 1+2+3) of length d and an atomic module M_1 (right side of Figs. 1+2+3):

- *Isolated*: As Fig. 2 shows, two modules of these types can be isolated such that there is no path that connect elements in M_d with elements in M_1. Since there is no connection, M_d cannot be extended by merging with M_1 in this case.
- *Connected*: Fig. 3 shows a module M_d that is connected with M_1 through the edge $e(a, b)$, where $a \in M_d$ and $b \in M_1$. When merging M_d with $e(a, b)$ such that the extended module M_{d+e} contains all elements in M_d and $e(a, b)$, then the length of M_{d+e} is $d + e = d + 1$ (length of M_d + length of $e(a, b)$). Merging M_d with the entire module M_1 yields M_{d+e+1} with length $d+1+1$ (length of M_d + length of $e(a, b)$ + length of M_1).
- *Overlap*: Fig. 4 shows the modules M_d and M_1 that are connected through a hub element a. In this case, all elements in M_1 are reachable from all elements in M_d through paths with length $d + 1$ (length of M_d + length of M_1). Therefore, merging the two modules results in a combined module with length $d + 1$.

Corresponding to these three cases, we have three merge operations that result in modules of different lengths, i.e. $M_d \cup e$, $M_d \cup e \cup M_1$ and $M_d \cup M_1$. Starting with atomic modules, we enumerate all possible merge operations, and in every iteration, apply only those ones, which result in modules that satisfy the length limit. Due to the scale-free nature of the ontology graph, we set the length limit to a small value, $d = 2$, in the implementation in order to obtain modules of manageable sizes.

7 Evaluation

We evaluate the approach based on real-world data made available through the NCBO BioPortal. It is a system supporting search and browsing over several document collections, using annotations that come from ontologies such as SNOMED. Annotations are stored in the NCBO Resource Index [9], and were mainly created using the NCBO Annotator [14]. The latter is basically a concept recognizer, which similar to entity linking based approaches, finds referent words in the documents, given concept names in the ontologies. It mainly employs syntactic concept recognition (using concept names and synonyms).

Data. As data, we use SNOMED Clinical as well as the documents that have been annotated with concepts from this ontology. In total, the data contain 45 thousands ontology nodes, 197 thousands documents and 8.3 million annotations between them that have been created previously using the NCBO Annotator and made available through the NCBO Resource Index.

Goals. The experiments we designed here aim at studying two questions: (1) How well does our technique for computing candidate annotations using coarse-grained topic-based representation compare against baseline approaches operating at the level of individual words (e.g. entity linking)? (2) What is the quality of the computed modules?

Metrics. To answer the first question, we look at the candidate annotation results. Notice that our topic-level approach can be seen as a blocking technique, which primarily, is a quick way to select relevant candidates. The analogy to blocking techniques is also here, scalability is a concern and thus the goal is to quickly and efficiently retrieve relevant candidates first, which are then refined subsequently. In data integration, outputs of blocking feed into more sophisticated algorithms that refine the results, while in this work, annotation candidates are used by experts for creating annotation manually. Due to this analogy, we use the same methodology that is applied for evaluating blocking techniques. Blocking aims at improving the efficiency, an aspect that is measured by the Reduction Ratio (RR), i.e. the number of pairwise element-level comparisons that can be reduced via blocking, compared to the baseline. At the same time, blocking should ensure that remaining candidates cover as many actual results as possible. In other words, blocking should preserve as many relevant results as possible, while filtering irrelevant ones. This is measured by Annotation Completeness (AC), which indicates how many results in the ground truth are included in the set of candidates. Since blocking should help reducing the search space of a more sophisticated baseline (that is used in the subsequent refinement process), its metrics are computed based on the results of that baseline, i.e. RR is computed based on the number of comparisons needed by the baseline and AC is derived from the results computed by the baseline. In our experiment, we aim to assess how our candidate annotation selection approach compares to the baseline as implemented by the NCBO Annotator, i.e. what is RR and AC based on the operations performed and annotations produced by that system. Let D be the document set to be annotated, G the ontology,

M the set of all candidate annotations produced by our approach, and M^* the annotations produced by the NCBO Annotator, we define the following metrics:

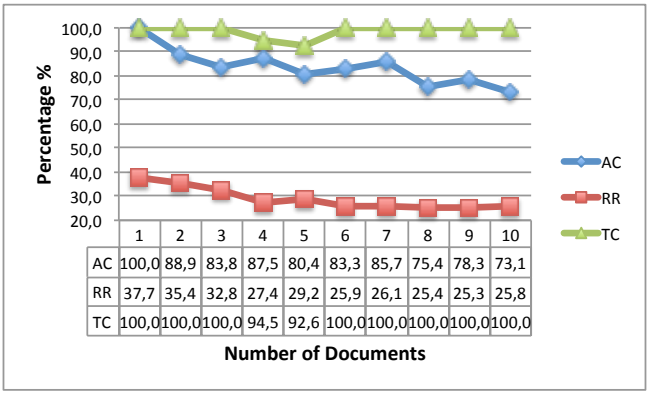
- *Annotation Completeness* (AC) measures the completeness of a generated module compared to baseline results. It is calculated as $AC = \frac{M \cap M^*}{M^*}$. The value of AC is in the interval $[0, 1]$, and higher value indicates higher effectiveness.
- *Reduction Ratio* (RR) measures the reduction in the number of pairwise comparisons, calculated as $PC = 1 - \frac{|M|}{|G|}$, where $|G|$ denotes the number of ontology nodes in G. This is because without blocking, all nodes in G have to be considered, instead of focusing on the candidates in M only. The value of RR is in the interval $[0, 1]$, and higher value indicates higher efficiency.

The aspect of module quality is more intricate, which as discussed, shall involve end user judgments. In this initial experiment, we perform the following procedure as an alternative mean to study this aspect in an automatic fashion: recall that our topic extraction technique rests on the assumption that contextual elements are more useful when they are on-topic. That is, on-topicness is used as a proxy of quality (based on that assumption). A module is of higher quality when its elements are about the same topics as the set of documents to be annotated. Due to the nature of the proposed extraction method, the on-topicness of a module M is optimized for the input documents D, which shall be annotated. However, using annotations in the Resource Index, we can select other subsets of documents to analyze the on-topicness of M. In particular, we select a collection D_{test}, where $D_{test} \cap D = \emptyset$, and documents $d \in D_{test}$ are annotated by elements in M (as indicated by annotations in the Resource Index). That is, according to the Resource Index, M is a good module for annotating D_{test} because it contains all its annotations. We study whether M is also good for D_{test} in the sense that D_{test} and context elements in M capture the same topics, using a measure of topic overlap, where a higher overlap between D_{test} and M indicates a higher "quality" of M:

- *Topic Completeness* (TC) measures the overlap of M and D_{test} as $TC = \frac{|Z_{D_{test}} \cap Z_D|}{|Z_D|}$, where $Z_{D_{test}}$ and Z_D are the topic sets computed for $d_{test} \in D_{test}$ and $d \in D$, respectively. The value of TC is in the interval $[0, 1]$, and higher value indicates higher effectiveness.

Setting. Every time, n documents are randomly chosen to form the set D of documents to be annotated. Then, we compute M, and based on the annotations in the Resource Index, we derive M^* and D_{test} to compute RR, AC and TC. The reported results represent an average of 20 runs. We performed the experiments using two parameter settings: (1) In the first setting, we aim to assess the sensitivity of the results with respect to the number of documents in the input set D, and in the second, we analyze the impact of k_{topics}.

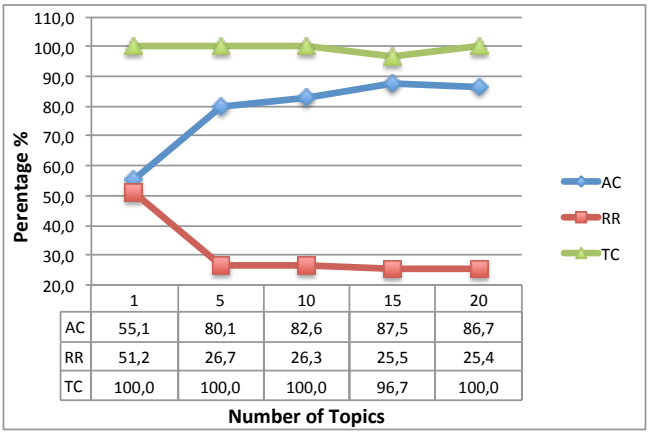
Results. First, we fix k_{topics} to 10, and vary the number of documents to be annotated, $|D|$, from 1 to 10. As shown in Fig. 5, AC and RR decrease as



	1	2	3	4	5	6	7	8	9	10
AC	100,0	88,9	83,8	87,5	80,4	83,3	85,7	75,4	78,3	73,1
RR	37,7	35,4	32,8	27,4	29,2	25,9	26,1	25,4	25,3	25,8
TC	100,0	100,0	100,0	94,5	92,6	100,0	100,0	100,0	100,0	100,0

Fig. 5. The Effect of Input Size $|D|$

$|D|$ increases. This is mainly because with the increase of documents in $|D|$, the number of topics corresponding to these documents also increases. However, if there are more than 10 relevant topics, the ones beyond the top-10 are not considered in this setting. Hence, relevant annotations that are associated only with these excluded topics cannot be found in this case. With the increase in relevant topics, the number of annotation candidates also increases, resulting in lower RR. However, RR remains stable when $|D| > 5$. This is because in these cases, the number of relevant topics is always more than 10. Since only a maximum of 10 topics are used, this setting in fact puts a limit on the number of candidates being considered. Fig. 5 shows that the effect of $|D|$ on TC is not obvious. Except for two outliers, TC is always 100 percent.



	1	5	10	15	20
AC	55,1	80,1	82,6	87,5	86,7
RR	51,2	26,7	26,3	25,5	25,4
TC	100,0	100,0	100,0	96,7	100,0

Fig. 6. The Effect of k_{topics}

Then, we fix the input size $|D|$ to 10, and analyze the sensitivity of results w.r.t. the number of topics k_{topics}. The results of this experiment are shown in Fig. 6. Clearly, when the number of topics increase, AC increases, but RR decreases. More topics means more annotation candidates can be taken into account, resulting in higher AC. This increase is sharp when k_{topics} changes from 1 to 5. But then, improvements become smaller as higher values are used for k_{topics}. AC even decreases when k_{topics} changes from 15 to 20. This means that while more candidates can be considered when k_{topics} is large, the number of irrelevant ones also increases. It seems that 15 is the right number of topics, which yields best results. A change of k_{topics} from 1 to 5 has also the largest effect on RR. After that, RR remains fairly stable. We found that TC is also not affected by the number of topics. Actually, the computed modules are larger in size when using a larger number of topics. When there are more topics, more annotation candidates are produced and used for module construction. However, this does not seem to have any effect on the on-topicness of the context elements included in the modules.

8 Conclusions

In this work, we elaborate on the new problem of ontology-based annotation support. Given ontologies (or any kind of graph-structured datasets), the idea is to extract modules from them containing annotation candidates as well as context elements that can help users in understanding the data and creating annotations manually. The underlying subproblems are related to the task of linking text elements to referent entities in structured datasets, and modularizing ontologies. We propose to use a topic-level representation to compute candidate links (candidate annotations) more efficiently, as well as a new strategy for ontology modularization based on "on-topicness". Initial experiments show promising results, while also suggesting that much more work is needed towards this direction. In particular, the usefulness of the computed modules can only be determined by experts, a question that has to be investigated thoroughly through user studies. Further, the techniques proposed for the computation of candidate annotations as well as for constructing modules represent only the first attempts towards solving this new problem. There is large room for improvement, both in terms of performance and result quality.

References

1. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *NIPS*, pages 601–608, 2001.
2. X. Chai, B.-Q. Vuong, A. Doan, and J. F. Naughton. Efficiently incorporating user feedback into information extraction and integration programs. In *SIGMOD Conference*, pages 87–100, 2009.
3. F. Chen, B. J. Gao, A. Doan, J. Yang, and R. Ramakrishnan. Optimizing complex extraction programs over evolving text data. In *SIGMOD Conference*, pages 321–334, 2009.

4. L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, and S. Vaithyanathan. Systemt: An algebraic approach to declarative information extraction. In *ACL*, pages 128–137, 2010.
5. L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *EMNLP*, pages 1002–1012, 2010.
6. P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. In *VLDB*, pages 399–410, 2007.
7. X. Han and L. Sun. A generative entity-mention model for linking entities with knowledge base. In *ACL*, pages 945–954, 2011.
8. X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *SIGIR*, pages 765–774, 2011.
9. C. Jonquet, P. LePendu, S. M. Falconer, A. Coulet, N. F. Noy, M. A. Musen, and N. H. Shah. NCBO Resource Index: Ontology-Based Search and Mining of Biomedical Resources. In *2010 Semantic Web Challenge*, 2010.
10. E. Meij, D. Trieschnigg, M. de Rijke, and W. Kraaij. Conceptual language models for domain-specific retrieval. *Information Processing & Management*, 46(4):448–469, 2010.
11. F. Reiss, S. Raghavan, R. Krishnamurthy, H. Zhu, and S. Vaithyanathan. An algebraic approach to rule-based information extraction. In *ICDE*, pages 933–942, 2008.
12. M. J. Samudrala R. A graph-theoretic algorithm for comparative modeling of protein structure. *J.Mol. Biol.*, 279:287–302, 1998.
13. S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *NIPS*, 2004.
14. N. H. Shah, N. Bhatia, C. Jonquet, D. L. Rubin, A. P. Chiang, and M. A. Musen. Comparison of concept recognizers for building the open biomedical annotator. *BMC Bioinformatics*, 10(S-9):14, 2009.
15. H. Simpson, S. Strassel, R. Parker, and P. McNamee. Wikipedia and the web of confusable entities: Experience from entity linking query creation for tac 2009 knowledge base population. In *LREC*, 2010.
16. H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. Springer-Verlag, Berlin, Heidelberg, 2009.
17. D. Trieschnigg, W. Kraaij, and M. J. Schuemie. Concept based document retrieval for genomics literature. In *TREC*, 2006.
18. X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.
19. S. E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina. Entity resolution with iterative blocking. In *SIGMOD Conference*, pages 219–232, 2009.
20. C. Zhai and J. D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM*, pages 403–410, 2001.