Detecting Meaningful Compounds in Complex Class Labels

Heiner Stuckenschmidt, Simone Paolo Ponzetto, and Christian Meilicke

Data and Web Science Group, University of Mannheim, Germany
{heiner,simone,christian}@informatik.uni-mannheim.de

Abstract. Real-world ontologies such as, for instance, those for the medical domain often represent highly specific, fine-grained concepts using complex labels that consist of a sequence of sublabels. In this paper, we investigate the problem of automatically detecting meaningful compounds in such complex class labels to support methods that require an automatic understanding of their meaning such as, for example, ontology matching, ontology learning and semantic search. We formulate compound identification as a supervised learning task and investigate a variety of heterogeneous features, including statistical (i.e., knowledge-lean) as well as knowledge-based, for the task at hand. Our classifiers are trained and evaluated using a manually annotated dataset consisting of about 300 complex labels taken from real-world ontologies, which we designed to provide a benchmarking gold standard for this task. Experimental results show that by using a combination of distributional and knowledge-based features we are able to reach an accuracy of more than 90% for compounds of length one and almost 80% for compounds of length two. Finally, we evaluate our method in an extrinsic experimental setting: this consists of a use case highlighting the benefits of using automatically identified compounds for the high-end semantic task of ontology matching.

1 Introduction

Conceptual models of information structures and information flows are a central concept in computer science. They play a crucial role in the design and maintenance of information systems. Besides the classical tasks of creating and evolving conceptual models, the task of identifying mappings between different models as a basis for integrating different systems has become more and more important. The problem of integrating different representations of reality is a long-standing problem in computer science. In particular, it is the core problem of the field of data integration. The database community has developed a variety of methods for identifying matching data elements both on the level of instance and schema data [7]. More recently, the problem of matching elements from different ontologies, namely formal models of an application domain, has been investigated in detail [36]. It has been argued that many matching techniques developed for schema matching can also be applied to ontology matching. However, questions

remain on whether further advances could be achieved by leveraging the formal semantics of ontologies.

Despite much research work in the field, existing approaches to ontology matching still have a number of limitations. For instance, almost all existing methods produce simple one-to-one matches between elements in the representations to be compared [10]. That is, most existing systems rely on the naive assumption that the representations to be compared represent reality at the very same level of granularity. A particular problem that can be observed when trying to match models that describe the domain at different levels of abstraction are situations where the class names describe complex constructs that do not have a direct counterpart in the other model, but their intended meaning can be expressed (or at least approximated) by a logical expression over simpler elements [37]. A complete solution to this problem amounts to developing novel, full-fledged methodologies to ontology matching that cover arbitrary one-to-many mappings. While we envision this as a longer-term goal requiring substantial research efforts, in this paper we provide a first step towards such a solution by addressing the problem of *understanding complex class labels*. More specifically, we focus on the task of *identifying meaningful compounds in complex ontology labels* that might refer to independent classes in a differently structured ontology. This is, to the best of our knowledge, the first attempt to address in detail this problem, which bears nevertheless a strong resemblance with other well-known tasks in Natural Language Processing and Information Retrieval – e.g., syntactic disambiguation of multiword expressions (also known as noun compound bracketing) [2] and query segmentation [3, 15, *inter alia*].

1.1 Problem Definition

Real-world ontologies, e.g., those providing semantic models of a highly specialized domain such as the medical one, often provide a description of their fine-grained concepts by means of complex labels that typically require some knowledge of the domain to make sense of. As an example, let us focus on the concept label *natural killer cell receptor 2B4*, which can be found in the Gene Ontology [1]. This label shows properties typical of complex ontology labels. Note that with 'complex' we refer here to the fact that the noun compound exhibits both syntactic and semantic ambiguity. That is, the label could be interpreted in different ways, depending on how its internal syntax is disambiguated. For instance, looking at the first four tokens of our example label, we see that there are at least three ways in which it could be bracketed, and thus interpreted:

(1) [natural killer] [cell receptor]
(2) [natural] [killer cell receptor]
(3) [natural killer cell] [receptor]

The first interpretation would be that the label describes the cell receptor of a natural killer. Clearly, this is for humans a quite implausible interpretation of the intended meaning of the label. Nevertheless, the two other possible bracketings

provide us with two equally plausible interpretations, which are both hard to rank as preferred interpretation, even by human subjects. The second possible interpretation, in fact, identifies the natural form of a killer cell receptor, whereas the third one the receptor of a natural killer cell – which is actually the correct interpretation, since 'natural killer cell' is a technical term in immunology. Note that, at a closer look, for semantic applications – such as, for instance, mapping label constituents (i.e., substring) to another resource – we need in practice a task formulation that goes beyond simple bracketing of adjacent noun phrases. First of all, meaningful parts of a label can actually overlap. In our example, these are 'natural killer cell' and 'cell receptor'. The term 'cell' is part of both components and links the two concepts to each other. Beyond that, there are also cases where meaningful compounds consist of terms that are not adjacent in the label. An example is the label 'British Crown colony' where all combinations of terms actually identify a meaningful concept: i) the 'British Crown', which is in charge of the colony, ii) 'Crown colony' indicating the property of the colony as belonging to a kingdom, and iii) 'British Colony', which describes that the colony is or was owned by great Britain.

To provide a workable problem definition, we define criteria for recognizing a meaningful compound within a complex label as follows:

Definition 1. *Given a complex concept label* $l = (l_1, \cdots, l_n)$ *a compound in* l *is a subsequence* $s = (s_1, \cdots, s_m)$ *of* l *where* $m < n$*. A compound* s *in* l *is meaningful if*

- s *is a grammatically correct noun phrase,*
- s *can be the label of a possible concept in some ontology,*
- s *retains a meaningful relation to* l*.*

Rather than providing a general or exhaustive solution, this definition is inspired by the intended application to ontology matching (Section 5). Since the ultimate objective is to find semantic relations to other ontologies, we are interested in parts of the label that can be found as concept labels in other ontologies (requirement 2). Clearly, we are only interested in those concepts that play some part in a complex mapping, and thus have some relation to the complex label (requirement 3). Admittedly, the definition is not unambiguous, so we must rely on human annotations as a reference (Section 4.1).

1.2 Contributions

In this paper, we investigate the problem of automatically detecting meaningful compounds in complex class labels as a first step towards complex ontology matching. The contributions of this paper are the following:

- We propose a supervised approach for recognizing meaningful compounds in complex ontology labels[1].

[1] In this work, we focus primarily on labels of length 3: however, our approach can be used in principle with labels of arbitrary lengths.

- We investigate different sets of statistical and knowledge-based features as a basis for the learning approach.
- We create a manually annotated benchmark dataset consisting of about 300 complex three-word labels taken from real-world ontologies.
- We show that, thanks to a combination of statistical and knowledge-based features, we can reach an accuracy of about 90% for compounds of length one and about 80% for compounds of length two.
- Based on the results of the experiments, we propose an unsupervised approach for detecting meaningful compounds in labels of arbitrary length.

2 Related Work

Label Analysis. Recently, there has been initial work addressing the analysis and use of complex labels for ontology enrichment and semantic matching. Manaf and others report results of a large scale analysis of the structure of class names on the Semantic Web [21]. They conclude that almost 90% of all class labels resp. identifiers on the semantic web are actually meaningful in that they provide a natural language description of the intended meaning of the class. More than 96% of these labels consists of more than one word. Further, they report that complex labels can be parsed syntactically as most labels use camel case syntax or special separators to delimit single words. In our previous work, we have used patterns over linguistic features generated through part-of-speech tagging, syntactic parsing and lexical semantic analysis to detect complex mappings between ontologies [30, 31]. In the area of business process modeling, Mendling and others have developed a method for analyzing activity labels based on different modeling styles observed in real world models [22, 20]. Other researchers focused instead on domain-specific resources ranging from biomedical ontologies like the Gene Ontology [11] and those found on BioPortal [27], all the way through identifiers found in source code [8].

Noun Phrase Chunking and Compound Bracketing. Two related problems from the field of Natural Language Processing (NLP) are text chunking (also referred to as shallow parsing) and noun compound bracketing. In contrast to full syntactic parsing, text chunking is concerned with the identification of flat, non-overlapping segments of a sentence which identify its basic non-recursive phrases corresponding to major parts-of-speech such as noun, verb and prepositional phrases. Noun phrase chunking is the special problem of identifying basic noun phrases within sentences. Due to the tight relation to full parsing, early approaches relied on established parsing methods [29]. Major advances were made thanks to the organization of a shared task as part of the Conference on Natural Language Learning in 2000 [33]. The participating systems reached an accuracy of over 90%, with the best performance being reported for a supervised approach based on Support Vector Machines [18]. Further advances were later achieved using better statistical approaches to tagging such as, for instance, Conditional

Random Fields [35]. While our task is similar to the chunking problem, identifying 'chunks' in class labels is much harder as labels typically do not have a regular grammatical structure. Similarly, meaningful compound identification is related to the other NLP task of noun compound bracketing, namely the syntactic disambiguation of multiword expressions [2]. For this task the best-performing models are based on a variety of different syntactic and semantic features [24, 39]. But while these contributions provide us with useful hints as to which kind of features we need for the task at hand (e.g., N-gram statistics), bracketing is primarily meant as a phrase-internal parsing task: that is, it does not cover cases of meaningful non-adjacent compounds.

Query Segmentation. A problem that is actually closer to our task is that of segmenting web search queries. Keyword queries, in fact, show similarities with class labels as they typically do not have a regular grammatical structure and are often composed of different meaningful compounds (e.g. 'New York budget hotels'). Bergsma and Wang showed that a combination of statistical and linguistic features can be used to learn optimal segmentations from examples with an accuracy ranging between 85% and 90% [3]. The results were obtained on a set of 1500 queries sampled from the AOL search query database, a corpus of more than 35 million queries. Zhang and others proposed an unsupervised approach that makes extensive use of background resources like WordNet and Wikipedia to detect potential segments, and applied it to the robust and ad-hoc tracks of TREC reporting good results [41]. However, due to the task-based evaluation approach they opted for, it is not possible to compare their results to the supervised approach of Bergsma and Wang. More recently, Hagen et al. have proposed in [14] a rather light-weight query segmentation method that mostly relies on N-gram statistics from the Google N-gram Corpus [4]. In follow-up work, they show that giving preference to segments that correspond to Wikipedia titles further improves the results [15]. The results reported by Hagen et al. are in the same range as the ones reported by Bergsma and Wang, thus showing that unsupervised approaches can also be competitive.

3 Learning to Detect Meaningful Compounds

We present a method for automatically determining meaningful compounds in complex class labels. Our approach builds upon existing techniques for query segmentation, which are, however, adapted to our specific problem. Following Bergsma and Wang, we propose a supervised approach, and focus in this first initial attempt to explore in detail the feature space for the task at hand.

3.1 Approach

Successful approaches to query segmentation detect segment boundaries based on different features of the neighboring words or, in the case of the unsupervised approach of Hagen et al. [14, 15], based on features of all words in the query. This

approach does not work for us, as we want to consider all word combinations in a complex label. We solve this problem by regarding each possible word combination as the binary decision problem of determining whether the respective word combination is a meaningful compound or not, and learn a decision function using supervised learning methods. That is, given a concept label, our task is to consider all proper subsequences and decide for each of them whether they are meaningful or not (along the lines of Definition 1). We train the classifier using a wide range of different features. While many features are taken directly from previous work on query segmentation, we go one step further by adding a number of new features more specifically targeted to capture the nature of ontology class labels. We finally arrived at a set of about 80 individual features from different categories, which we now turn to describe in detail.

3.2 Features

Statistical Features Building on the results of Hagen et al. that show the benefits of N-gram statistics for query segmentation, we use statistical features from large corpora, more specifically the N-gram-based scores for segments (same as proposed by Hagen et al.), as well as features capturing the distributional similarity and relations between words occurring within a label.

Features based on N-gram Statistics In [14] the authors propose a measure to estimate the quality of a complete segmentation of a keyword query based on the number of occurrences of a possible segment, normalized by the length of the segment (to account for the power law distribution of N-grams on the web):

$$score(S) = \sum_{s \in S, |s| \geq 2} |s|^{|s|} \cdot count(s)$$

Here S is the complete segmentation consisting of individual segments $s \in S$. Thus the score of a segment is given by $|s|^{|s|} \times count(s)$ where $count(s)$ is the number of occurrences in the N-gram corpus. We use this segment score for all possible word combinations in a class label as feature. Since Hagen et al. treat query segmentation as a global optimization problem, they implicitly consider the relation between the scores of different segments. In order to take this relation into account, we also use the quotient of the scores of all possible compounds as features. We use the Google N-gram corpus [4] to collect statistics for all N-grams up to length 5 and the jWeb1T API [13] to determine their frequency. In [15] the authors show that treating segments that correspond to Wikipedia titles differently improves the results. In the present work, we use empirical evidence from Wikipedia titles as a separate feature (see below), rather than integrating them directly into the N-gram score.

Features based on Word Similarity N-gram statistics crucially rely on counting the occurrence of the exact string making up the compound label in very large, i.e., Web-scale corpora. This way, *bank account* is a likely compound, as it frequently occurs in text. However, this is not able to capture that, for instance,

bank and *account* are strongly associated with each other since they also frequently occur in context, albeit not necessarily in adjacent order – e.g., as in '*open an account in a bank*'. Accordingly, we propose to relax the requirement of exact matching and turn to distributional semantic [38] as a way to estimate the degree of association between each of the compounds' constituents. For each segment s of a complex concept label (of length two), we accordingly compute the pairwise similarity between its tokens. To this end, we use DISCO [17], a freely available toolkit to build semantic spaces from text and compute distributional similarity. In this work, we use both first-order and second-order context vectors [34] to compute the semantic similarity between the compounds' tokens, and use these two similarity scores directly as features for the classifier.

Features based on Relation Extraction Open Information Extraction systems such as ReVerb [9] offer another rich source of information to compute the degree of relatedness between the constituents of a compound. Accordingly, we used the ReVerb dataset[2] to compute such a score based on the extraction of relations between sublabels. Given two sublabels, we query for all those triples where one appears in subject position and the other as object, and vice versa. We then count the number of distinct relations that appear in the resulting set of triples in the predicate position, and use this as feature for our classifier. Note that this provides us with an IE-based relatedness score that, in contrast to distributional similarity features, takes explicitly into account the context in which two constituents occur.

Resource-Based Features Previous work on query segmentation has shown that background knowledge from linguistic resources can significantly improve the identification of meaningful segments. We therefore also include a number of features based on available resources. Following the approach of Zhang and Hagen, we include WordNet and Wikipedia-based features. Since we are concerned here with ontology labels, we also add new, previously unexplored features that are based on the occurrence of words and compounds in the labels of classes, instances and relations of ontologies found on the semantic web.

Wikipedia-based Features Successful unsupervised approaches to query segmentation make use of Wikipedia to determine segments that correspond to meaningful concepts. We adopt this approach and test whether combinations of words from a concept label, including the complete label, correspond to a title of a Wikipedia page. The wide coverage of Wikipedia and the fact that Wikipedia pages are created by human editors and are subject to an intellectual revision process make it a very useful source of information about descriptions of meaningful concepts [16]. In order to determine whether a sequence of words corresponds to a Wikipedia title, we use the JWPL Wikipedia API [40].

[2] `http://openie.cs.washington.edu/`

WordNet-based Features WordNet was used as a dictionary in [41] to check whether a word in a query is a proper noun. We adopt and extend this idea. In particular, for each word in a class label, we collect all parts-of-speech (PoS) – namely any of noun, verb, adjective or adverb – it can have in WordNet. We consider PoS other than nouns to capture context-specific ambiguity across PoS – e.g., 'light' used as an adjective as in 'light armored vehicle'. We do not attempt at determining the unique exact PoS of the word in context, e.g., using a syntactic parser, as these typically perform badly when applied to small concept labels [26]. PoS of WordNet terms are retrieved using the JWNL API[3].

Ontology-based Features We additionally define a set of new, previously unexplored features that are more directly related to the nature of our task. Since, in our case, a meaningful compound consists of a phrase that could appear as a concept name within an ontology, we test for all words in a label whether they occur as the description of an element in existing ontologies available on the Semantic Web. Similar to the case of PoS in WordNet, we do not restrict the search to class names, but also test whether the phrase is used in the descriptions of relations or instances, since this make the candidate less likely to be a meaningful class name. This can be seen as an ontological version of the WordNet-based features described above. Further, for each pair of words in a label, we count the number of ontologies both words occur in. This can be seen as an ontological version of computing word co-occurrence. We use the Watson search engine for ontologies [6] as a tool for accessing available ontologies on the web and computing our features. This approach was inspired by [32], where the authors use Watson as a mechanism to detect background knowledge for ontology matching.

Using Classification Results within a Bootstrapping Architecture The different classification tasks that originate from a single label are not independent of each other. Consequently, we first classify shorter compounds and then use the predicted class and the confidence of the classifier for compounds of length n as additional features for classifying compounds of length $n+1$. To this end, we first train base classifiers to decide whether the individual tokens of the label, say 'British Crown Colony' - (referred to as (A) British, (B) Crown and (C) Colony) are meaningful terms on their own. In the next step, the class labels and confidence values of these classifiers are used as features for classifiers that decide whether two-word combinations – i.e., British Crown (AB), Crown Colony (BC) and British Colony (AC), in our case – are meaningful labels themselves.

4 Experiments

4.1 Gold-standard Dataset

To create a gold-standard for training and evaluating our classifiers, we used the **S**uggested **U**pper **M**erged **O**ntology (SUMO) [25]. SUMO, and its domain

[3] `http://sourceforge.net/projects/jwordnet/`

ontologies, form a large formal ontology used for research and applications in search, linguistics and reasoning. SUMO contains concepts that describe the world on a very abstract level, while some of the integrated ontologies cover very specific topics like communication or transportation – the latter, for instance, distinguishing between different types of cargo ships[4].

Analysis of the concept labels found within SUMO revealed that 1579 concepts are described by non-compound labels, whereas 1755 concepts have two-word labels, 635 have three-word labels, and 236 are described using concept labels made up of more than three words. From the whole set of three-word compounds we randomly sampled a subset of 300 labels. These labels cover completely different topics, and range across domains as diverse as from military (e.g., *amphibious assault vehicle*) to medical ones (e.g., *yellow fever virus*)[5].

Given a concept label of the form ABC, three human judges were asked to provide a ground truth by annotating the label's compounds, namely any of A, B, C, AB, BC, or AC, as meaningful or not, based on Definition 1. The final gold standard was created by aggregating the single annotators' judgments based on majority voting. In order to quantify the quality of the annotations and the difficulty of the task we computed the inter-annotator agreement using the kappa coefficient [5] – we use Fleiss' kappa [12]. Our annotators achieved an agreement coefficient κ of .73, .70 and .60 for annotating the two-word compounds AB, BC and AC, respectively. An average agreement of $\kappa = .68$ indicates substantial agreement between annotators, thus corroborating the overall quality of the annotated data, as well as the well-definedness of our task.

4.2 Experimental Setting

We perform experiments using the Rapidminer toolkit [23], version 5. We set up two learning processes: i) one for classifying single words that uses solely external features of words and word combinations, and ii) a second one for classifying two-word segments that uses the results of classifying single words, together with external features. For both tasks, we experimented with a number of different learning algorithms. Below, we report results using Support Vector Machines (SVM) and Neural Networks (NN), since these methods showed a significantly better performance than other methods. We use SVM with dot product kernels and NNs with one hidden layer (additional parameters can be found in the process definitions).

Many of our features (e.g., distributional similarity) can be only computed pairwise between different words, and thus require multi-word compounds. Accordingly, we conducted a finer-grained feature analysis using two-word combinations only: in this setting, statistical and knowledge-based features were evaluated separately, in order to quantify the different contribution of background knowledge vs. statistics from large corpora for our task. Given the

[4] SUMO is originally published in the SUMO-KIF format [25]. In our work we use the OWL version available at `http://www.ontologyportal.org/`.

[5] The gold standard is freely available at `https://madata.bib.uni-mannheim.de/57/`.

feature type	*statistical*		*knowledge-based*		*all*	
learning algorithm	SVM	NN	SVM	NN	SVM	NN
A					87.91	**91.21**
B					**90.11**	87.91
C					94.51	**98.90**
average					90.84	**92.67**
AB	74.34	79.96	79.28	79.95	79.27	**80.27**
BC	70.04	74.35	69.70	**81.84**	80.60	79.27
AC	65.78	63.13	**75.96**	71.37	75.30	74.03
average	70.05	72.48	74.99	77.72	**78.39**	77.86

Table 1. Results on the identification of meaningful compounds. Performance figures for AB, BC and AC are obtained using the bootstrapping architecture described in Section 3, and thus use classification results for A, B and C as additional features.

limited size of our dataset, we employ ten-fold cross validation for all our experiments. For evaluation, we use standard measures of recall, precision and accuracy: below, we only report accuracy for each classification task for the sake of brevity. However, all detailed results for our experiments, the Rapidminer processes, and the full feature tables can be found online at `https://madata.bib.uni-mannheim.de/57/`.

4.3 Results

We present our results in Table 1, where we report accuracy figures for the detection of meaningful, single-word compounds (i.e., A, B or C), as well as two-words (namely, any of AB, BC or AC). Overall, our results for the classification of single-word compounds are generally favorable, with performance figures on average $> 90\%$ for both SVMs and neural networks. When looking at the performance on each single token position, we notice the higher results on the rightmost word, namely C: this is because this generally corresponds to the lexical head of the noun phrase[6]. These constituents typically identify, from a semantic point of view, the concept's super-concept, e.g., *amphibious assault vehicles* are *vehicles* (cf. also the head-matching heuristics from [26]) and provide a meaningful concept label in the vast majority of cases. Results on A and B, in contrast, are lower since these tokens are meaningful in a smaller number of cases, which crucially depends on a variety of complex factors, ranging from syntactic – like, for instance, the token having a PoS other than noun (e.g., an adjective, as in "merchant *marine* ship") – through semantic – for example, the single constituent having no meaning related to that of the overall phrase, as in "rift *valley* fever").

Results on the classification of two-word constituents are lower in that these instances also require in many cases complex decisions integrating heterogeneous

[6] The head of a phrase is the word which is grammatically most important in the phrase, since it determines the nature of the overall phrase [28]. For basic non-recursive noun phrases, this typically corresponds to the rightmost noun.

features. In general, we note that results on AC are lower that those on AB or BC, which is in line with the higher difficulty of the task highlighted by the lower inter-annotator agreement of our human raters (Section 4.1). When looking at the contribution of each single feature group, we note that, in general, knowledge-based features tend to perform better than statistical ones. This is because, while statistical information provides us with better coverage, knowledge-based features are indeed superior for the present task in that they rely on very large amounts of human supervision from large-scale, high-quality semantic resources like Watson, WordNet and Wikipedia. However, the complementarity of both feature types is shown by the overall results – namely those obtained by averaging performance over AB, BC and AC – being obtained when using both statistical and knowledge-based features. We take this to be good news, since it suggests that better performance on this task can be achieved in the future by exploring other heterogeneous knowledge sources, as well as their combination with robust learning algorithms.

5 Use Case

We next analyze whether the detection of meaningful compounds provides us with a valuable knowledge source for the task of matching complex ontology labels. A complete solution for the mapping task itself is beyond the scope of this paper: however, in this work we can already report about some experiments that yield relevant insights. Given a complex compound label, we first apply our method to segment the labels into meaningful parts. We then try to detect a concept with an equivalent or highly similar meaning within a target ontology. Our hunch here is that robust performance on this simplified task indicates that we can use use the results of our segmentation as input to generate partial mappings, which are later used to solve the complex matching task as a whole.

In the following we make use of the same dataset described in Section 4.1. Since there exists no evaluation dataset that deals with the problem of complex ontology matching, we formulate a pseudo-matching task as follows. For each compound label from our dataset we remove the corresponding concept from the SUMO ontology. Then we try to anchor this concept back within the target resource. This simulates the task of mapping a concept to an ontology, where an equivalent concept does not exist as named concept. In such a scenario, the concept can be anchored at the right position in the concept hierarchy, or it might be possible to construct an equivalent complex concept description. Let C denote such an concept, let $l(C)$ denote its label, and let $l_m(C) = \{m_1, ..., m_n\}$ denote the set of compounds that have been annotated as meaningful (either from our system or from the human annotators). In our experiments we then aim at creating a mapping for each m_i to one of the concepts in SUMO. In particular, we create a mapping if we find a concept D with $l(D) \cong m_i$, where $\cong$ refers to string equality after normalization. The results of the Ontology Alignment Evaluation Initiative have shown that this approach results in highly precise mappings that are often hard to beat in terms of F-measure [10].

	baseline	learning algorithm	gold standard
precision	20.1	33.1	31.6
recall	100	91.6	93.2
F-Measure	33.5	48.6	47.2

Table 2. Mapping fragments of a compound label to concepts.

In Table 2 we report on the fraction of labels from $l_m(C)$ that can be matched to a concept in SUMO. Performance is computed using standard metrics of precision (fraction of all labels for which a mapping has been generated), recall (fraction of generated mappings compared to the mappings generated by taking all possible sublabels into account) and F_1-measure (the harmonic mean of precision and recall). We compute these scores in three different settings, namely for: (1) a baseline that considers all sublabels as meaningful combinations; (2) the output of our best-performing supervised classifier from Section 4.3; (3) the gold standard provided by human annotators (Section 4.1), which theoretically provides us with an upper bound for this task. Taking all sublabels into account, we achieve a recall of 100% (by definition) and a precision of 20.1%. Using the output of our algorithm yields instead an increased precision of 33.1%, while maintain recall above 90%. Overall, we can increase the F-measure from 33.5% to 48.6%: we take these as good results with respect the second bullet point in Definition 1 ('it must be a possible concept in some ontology'). Finally, we note that precision and recall change only to a very limited degree when compared against the results of using the gold-standard labels, thus indicating the overall robustness of our approach.

We next analyzed how many mappings generated during our experiments led to a concept that is a superclass of C. This happens for 58.9% of the instances in the dataset, regardless of whether we use automatically-detected compounds or gold-standard labels. Due to the artificial nature of these experiments – which merely consisting of removing a concept from its place in the reference ontology, as opposed to the full-fledged ontology matching task – we can easily compute these figures in our experimental setting. However, note that in a real matching scenario it is a challenging task to find the right position in the concept hierarchy for a given complex concept label. While in our use case ≈60% of the generated mappings help us solve the task of attaching the concept to the right place in the target concept hierarchy, the remaining ≈40% of the mappings establish links to other concepts. Error analysis revealed that these 40% do not necessarily consist of incorrect mappings. Quite contrary, they might be required to construct complex concept expressions. An example is the concept *fish carrier ship*. The concept *ship* is a superclass of the concept, while the concept *fish* is located in a different branch of the concept hierarchy. A correct mapping would express that a *fish carrier ship* is a *ship* that *carries* the cargo *fish*. That is, this example illustrates the task that needs to be solved for constructing precise equivalence mappings to complex concept descriptions.

6 Conclusions and Future Work

In this paper we presented an approach to detect meaningful compounds within complex ontology class labels. We proposed to view this as a binary classification task, and used a supervised classifier to explore a wide variety of features for solving this problem. Our results indicate that similarly, for instance, to previous results in query segmentation, supervised learning methods offer a viable solution for our task. In particular, they provided us with a complete framework to test many different features and accordingly understand the role and benefits of different knowledge sources. Our best results are obtained by combining statistical and knowledge-rich features, and indicate that future advances could be obtained by additional work on the feature engineering side.

We additionally evaluated the output of our classifier as source for a pseudo ontology matching task with complex class labels. The results indicate that we have to distinguish between two main objectives, in order to solve the challenging problem of matching compound labels. First, we need to identify a concept in the target ontology that is more general than the concept we want to match. So far, we can use our algorithm for detecting meaningful compounds: however, our algorithm cannot determine which of these compounds corresponds to a more general class, i.e., which of the constituents is the head noun. With this additional information we would be able to generate mappings expressing a subsumption relation. Extending our method to detect head nouns would thus be highly beneficial for generating correct subsumption mappings. Second, we have to aim at the construction of complex concept descriptions that are equivalent to the concept denoted by the compound label. This task is obviously much harder than the previously mentioned task. Let us focus again on the example *fish carrier ship* from the previous section. Constructing the equivalent concept description requires more knowledge than identifying the head noun. Moreover, we need to understand which relations hold between those sublabels that have been annotated to be meaningful. For this, relation extraction (which we merely used as a feature in this work) and semantic parsing [19] could prove useful.

With this work we aim at providing a first step towards understanding and solving the problem of matching complex concepts labels. The first results are promising in that our experiments helped us better understand the next steps that need to be taken into account for solving the concrete matching problem. Future work will focus on the open challenge of generating mappings for concepts labeled by compound expression. For generating equivalence mappings, we will turn to analyzing the relation between meaningful sublabels, in order to find an isomorphism between the structures on the linguistic layer and the structures that can be constructed by building complex concept descriptions.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M.,

Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature genetics 25(1), 25–29 (2000)
2. Baldwin, T., Kim, S.N.: Multiword expressions. In: Indurkhya, N., Damerau, F.J. (eds.) Handbook of Natural Language Processing, 2nd Edition. CRC Press, Taylor and Francis Group (2010)
3. Bergsma, S., Wang, Q.: Learning noun phrase query segmentation. In: Proc. of EMNLP-CoNLL-07. pp. 819–826 (2007)
4. Brants, T., Franz, A.: Web 1T 5-gram version 1. LDC2006T13, Philadelphia, Penn.: Linguistic Data Consortium (2006)
5. Carletta, J.: Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics 22(2), 249–254 (1996)
6. d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D.: Towards a new generation of semantic web applications. IEEE Intelligent Systems 23(3) (May/June 2008)
7. Doan, A., Halevy, A.: Semantic-integration research in the database community. AI Magazine 26(1), 83–94 (March 2005)
8. Enslen, E., Hill, E., Pollock, L., Vijay-Shanker, K.: Mining source code to automatically split identifiers for software analysis. In: Proc. of MSR-09. pp. 71–80 (2009)
9. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam: Open information extraction: The second generation. In: Proc. of IJCAI-11. pp. 3–10. AAAI Press (2011)
10. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology alignment evaluation initiative: Six years of experience. Journal on Data Semantics XV, 158–192 (2011)
11. Fernandez-Breis, J.T., Iannone, L., Palmisano, I., Rector, A.L., Stevens, R.: Enriching the gene ontology via the dissection of labels using the ontology pre-processor language. In: Proc. of EKAW-10. pp. 59–73 (2010)
12. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychological bulletin 76(5), 378 (1971)
13. Giuliano, C., Gliozzo, A., Strapparava, C.: FBK-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In: Proc. of SemEval-2007 (2007)
14. Hagen, M., Potthast, M., Stein, B., Brutigam, C.: The power of naive query segmentation. In: Crestani, F., Marchand-Maillet, S., Chen, H.H., Efthimiadis, E., Savoy, J. (eds.) Proc. of SIGIR-10. pp. 797–798 (2010)
15. Hagen, M., Potthast, M., Stein, B., Brutigam, C.: Query segmentation revisited. In: Proc. of WWW-11. pp. 97–106 (2011)
16. Hovy, E., Navigli, R., Ponzetto, S.P.: Collaboratively built semi-structured content and Artificial Intelligence: The story so far. Artificial Intelligence 194, 2–27 (2013)
17. Kolb., P.: DISCO: A multilingual database of distributionally similar words. In: Proc. of KONVENS-08 (2008)
18. Kudoh, T., Matsumoto, Y.: Chunking with support vector machines. In: Proc. of NAACL-01. pp. 1–8 (2001)
19. Kwiatkowski, T., Choi, E., Artzi, Y., Zettlemoyer, L.: Scaling semantic parsers with on-the-fly ontology matching. In: Proc. of EMNLP-13. pp. 1545–1556 (2013)
20. Leopold, H., Smirnov, S., Mendling, J.: Recognising activity labeling styles in business process models. Enterprise Modelling and Information Systems Architectures 6(1), 16–29 (2011)
21. Manaf, N.A.A., Bechhofer, S., Stevens, R.: A survey of identifiers and labels in OWL ontologies. In: Proc. of OWLED 2010 (2010)

22. Mendling, J., Reijers, H., Recker, J.: Activity labeling in process modeling: Empirical insights and recommendations. Information Systems 35(4), 467–482 (2010)
23. Mierswa, I.: Rapid miner. Künstliche Intelligenz 23(2) (2009)
24. Nakov, P., Hearst, M.: Search engine statistics beyond the n-gram: Application to noun compound bracketing. In: Proc. of CoNLL-05. pp. 17–24 (2005)
25. Pease, A.: Ontology: A Practical Guide. Articulate Software Press, Angwin, CA (2011)
26. Ponzetto, S.P., Strube, M.: Taxonomy induction based on a collaboratively built knowledge repository. Artificial Intelligence 175, 1737–1756 (2011)
27. Quesada-Martínez, M., Fernández-Breis, J.T., Stevens, R.: Lexical characterization and analysis of the BioPortal ontologies. In: Proc. of AIME-13. pp. 206–215 (2013)
28. Radford, A.: Syntax: A minimalist introduction. Cambridge University Press, Cambridge U.K. (1997)
29. Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In: Proceedings of the Third Workshop on Very Large Corpora. pp. 82–94 (1995)
30. Ritze, D., Meilicke, C., Šváb Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: Proc. of OM-2009 (2009)
31. Ritze, D., Völker, J., Meilicke, C., Šváb Zamazal, O.: Linguistic analysis for complex ontology matching. In: Proc. of OM-2010 (2010)
32. Sabou, M., d'Aquin, M., Motta, E.: Exploring the semantic web as background knowledge for ontology matching. Journal on Data Semantics 11, 156–190 (2000)
33. Sang, E., Buchholz, S.: Introduction to the CoNLL 2000 shared task: Chunking. In: Proc. of CoNLL-00. pp. 127–132 (2000)
34. Schütze, H.: Automatic word sense discrimination. Computational Linguistics 24(1), 97–124 (1998)
35. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proc. of HLT-NAACL-03. pp. 134–141 (2003)
36. Shvaiko, P., Euzenat, J.: Ontology matching: : state of the art and future challenges. IEEE Transactions on knowledge and data engineering 25, 158–176 (2012)
37. Stuckenschmidt, H., Predoiu, L., Meilicke, C.: Learning complex ontology mappings - a challenge for ILP research. In: Proc. of ILP-08 - Late breaking Papers (2008)
38. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)
39. Vadas, D., Curran, J.R.: Parsing noun phrase structure with CCG. In: Proc. of ACL-08: HLT. pp. 335–343 (2008)
40. Zesch, T., Müller, C., Gurevych, I.: Extracting lexical semantic knowledge from Wikipedia and Wiktionary. In: Proc. of LREC-08 (2008)
41. Zhang, W., Liu, S., Yu, C., Sun, C., Liu, F., Meng, W.: Recognition and classification of noun phrases in queries for effective retrieval. In: Proc. of CIKM-07. pp. 711–720 (2007)