

About the Influence of Negative Context

Nadine Steinmetz

Hasso Plattner Institute for Software Systems Engineering
Potsdam, Germany

Email: nadine.steinmetz@hpi.uni-potsdam.de

Harald Sack

Hasso Plattner Institute for Software Systems Engineering
Potsdam, Germany

Email: harald.sack@hpi.uni-potsdam.de

Abstract—Semantic analysis extracts semantic information from natural language texts and endeavors to make implicit facts explicit. Context and experience – in terms of previously achieved knowledge – are essential to solve this task. Confident semantic information from ambiguous natural language can only be obtained if set in a sufficient context. Conventional Named Entity Mapping algorithms use context as positive example environment for the disambiguation process. Traditional machine learning algorithms also apply negative examples to train a classifier for a specific subject. For Named Entity Mapping this can trivially be achieved by manual curation of black lists. These black lists contain entities that do not make sense in the given context. This paper describes an approach how to achieve a negative context dynamically during the disambiguation process and how to make use of this negative context for subsequent analysis steps.

I. INTRODUCTION

Recently, semantic analysis algorithms have become an essential part of (multimedia) document management systems [1]. Traditional content management systems are enhanced by semantic services including Named Entity Mapping (NEM) algorithms, i.e. mapping of a unique semantic entity to an ambiguous natural language expression¹, to enable semantic tagging, search, and retrieval (cf. Apache Stanbol²). Known NEM approaches, as e.g. *DBpedia Spotlight*³, or *The Wiki Machine*⁴ use available knowledge about semantic entities to enable unique identification and mapping in various contexts. The knowledge applied for disambiguation and mapping is mostly collected and derived from knowledge bases or large text corpora such as the online encyclopedia Wikipedia⁵. To enable a unique disambiguation and mapping of a natural language term, additional information about its context is required, which usually is derived from enclosing natural language text or further text-based metadata.

Current NEM approaches make use of co-occurrences of the entities under consideration within the utilized document corpora or exploit property links and page links within the knowledge base. Based on this *positive context* the identification and mapping process from ambiguous natural language terms to unique semantic entities is performed. Usually, NEM approaches benefit from the fact that particular entities are

emphasized among all remaining entity candidates, because they maintain a specific relationship with the positive context.

To the best of our knowledge, current NEM approaches do not take advantage of eliminating potential entity candidates, because they *do not* make sense with respect to a given context. A standard but rather limited approach to exclude non meaningful or wrong entity mappings is the maintenance of black lists. Black lists are usually manually generated lists of semantic entities that should not be considered for analysis or mapping because they do not make sense wrt. the given general context of the application. Manual black listing is carried out by human experts who are familiar with the application context and often does require high manual effort. Therefore, the automatic generation, maintenance and application of dynamic black lists in terms of a *negative context* would be rather helpful.

Similarly to a positive context that affirms given assumptions, a negative context can be defined. Semantic entities can either be explicitly member of a black list or they are closely related to entities within this negative context. As a starting point for the disambiguation process on a natural language text as a whole, the expectedly most reliable entity mapping might be chosen, where all discarded negative candidate entities, i.e. candidate entities of the wrong meaning according to the given context, will form the basis of the negative context. For further disambiguation and entity mapping, if there is a relationship among a new candidate entity and the negative context, the respective entity will be penalized and devalued. For the disambiguation of semantic entities within a natural language text, already (reliably) disambiguated entities of a sentence or a paragraph will form the positive context, while discarded candidate entities wrt. the text under consideration constitutes the negative context.

This paper presents an approach how to dynamically generate negative context to support the disambiguation of natural language text and text-based metadata and how to apply this negative context to eliminate potentially wrong entity mappings from the candidate entities to obtain a more precise and reliable disambiguation.

The paper is organized as follows: Section II lists known approaches for the application of negative context and how the term is also used in other research areas. Our approach of creating the negative context and applying it for the disambiguation process is described in Section III. An evaluation of the proposed method has been performed on two different

¹often also referred to as Named Entity Linking or Disambiguation

²<http://stanbol.apache.org/>

³<http://dbpedia-spotlight.github.io/demo/>

⁴<http://www.machinelinking.com/wp/demo/>

⁵<http://www.wikipedia.org>

datasets. The results and the discussion of the results is presented in Section IV. Section V concludes the paper and gives an overview on ongoing and future work.

II. RELATED WORK

To the best of our knowledge there exists no other NEM approach that takes a negative context into account to disambiguate ambiguous terms within a text. However, context and also negative context are concepts used in other research fields, such as linguistics, psychology or sentiment analysis. Following, we briefly sum up the efforts of other research fields dealing with negative contexts. Additionally, we also refer to other NEM approaches in comparison to the one proposed in this paper.

Context can be considered as the sum of available information items that put together enable unambiguous determination of the meaning of information [2]. Natural language processing (NLP) defines so-called negative and positive polarity items that bind a specific context. Negative polarity items (NPI) are lexical elements that occur in negative contexts only, as e.g., the term 'any', while positive polarity items, on the other hand, are in general excluded from negative contexts, as e.g., the term 'some'. Thus, the definition of a negative context depends on the language specific characteristics and designates contexts that license the occurrence of undisputed NPIs [3].

Negative emotional contexts denote situations, where people feel unhappy with the result of an event or other occurrences[4]. Sentiment analysis deals with the problem of negation or agreement within natural language to find out positive or negative influences in a context [5]. This paper does not cover negative context in terms of sentiments. Moreover, the term *negative context* is considered as opposite to the positive context applied to disambiguate ambiguous terms.

Deep or conceptual semantic analysis of natural language also addresses negative context effects [6]. Various contextual situations influence the comprehension of ambiguity in natural language. McCrae describes cross-modal contexts to classify different levels of cognitive input. Our work raises the problem of different contexts to the Linked Data level and addresses negative context in terms of NEM approaches.

NEM algorithms either try to identify the named entities in a natural language text by using trained classifiers [7] or the application of analytical methods [8]. Both approaches make use of the context, where the term to be disambiguated is embedded in, to identify relationships among terms or entities. This kind of context can be considered as *positive context*, because it is used to support previously made assumptions. Machine learning typically uses positive but also negative training samples to learn a model that best separates positive from negative samples for the given concept. [9].

For purely analytical approaches this cannot be achieved in first place. *Negative context* states which entities are not relevant for the given context, i.e. which interpretation might be discarded wrt. the current context. Usually, after successful disambiguation entities belonging to the negative context can easily be identified.

This paper presents an approach how to build negative context for an analytical NEM algorithm dynamically during the analysis process and how to make use of it.

III. USING NEGATIVE CONTEXT

In general, an NEM algorithm comprises three different steps:

- 1) Recognition of Named Entities
- 2) Detection of potential entity candidates in the given knowledge base
- 3) Disambiguation of the named entity and selecting the best matching candidate

The first step detects prospective named entities in the given text. In step two, for the textual representations potential entity candidates are retrieved in the given knowledge base for every previously detected named entity. For this step labels and alternative labels are collected to be able to search within multiple textual representations of the semantic entities of the underlying knowledge base. Subsequently the named entities that have multiple entity candidates assigned have to be disambiguated by using the given context. The workflow is also depicted in Figure 1.

In the following section the disambiguation process of named entities is described in general. Afterwards, the context model used for the disambiguation of terms in heterogenous contexts is described. The context model and the calculation of the confidence value is described in detail in [10]. This context model evaluates terms according to determined characteristics and brings them in a certain order respecting the achieved scores. This approach enables a disambiguation beginning with the prospectively most correct term within a context. It also facilitates the construction of a negative context. This procedure is described in section III-C. The adapted approach of the disambiguation process using the negative context is depicted in section III-D.

A. Disambiguation Process in General

During the disambiguation process all candidates for a named entity are scored according to the given context. Usually, the scoring method is an *additive* process. Thus, the total score for an entity candidate is added up from all relationships the entity candidate has within the context. The more relationships an entity candidate has to the context and the closer these relationships are⁶ the higher the total score the candidate receives. The entity candidate that achieves the highest score is chosen as prospectively correct disambiguation.

This type of disambiguation process takes into account only *positive context* information. The more evidences within this positive context the higher the score. In this paper we present an approach that additionally to the additive methods also considers subtractive factors on the score of an entity candidate. To receive evidences that affect the disambiguation score negatively a *negative context* has to be constructed.

⁶respectively the higher the co-occurrence of the entity and the terms of the context based on a common corpus

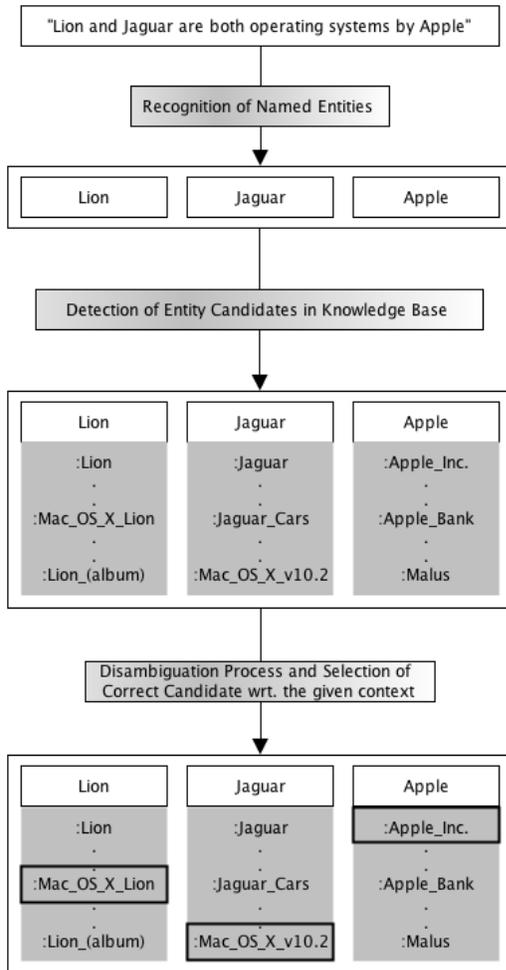


Fig. 1. The NEM process in an overview

This means, facts that lead to a negative scoring of an entity candidate need to be provided.

In the most simple way, such negative facts can be derived from entity black lists. These black lists can be composed of instances of ontology classes, categories, or topics that should not be taken into account for entity mapping. For example, in a text about “space shuttles” entities assigned to the topic “Pop music” most probably can be disregarded. Such black lists of topics, ontology classes or categories are often created manually and applied on documents where the content and its topic is known before the disambiguation process starts. Unfortunately, this is a requirement which usually can not be fulfilled when processing random web documents, as e.g. videos or simple texts. The entire knowledge base has to be taken into account when named entities are mapped to semantic entities. Therefore, we have developed an approach to construct a negative context successively during disambiguation. Our approach is based on the assumption that the negative context is built up by using excluded and eliminated entity candidates from the current disambiguation

process. Entities that have been an entity candidate for a natural language term in the context, but did not achieve any positive score in the disambiguation process, thus can be considered as not relevant within the current context. These entities will successively be added to the negative context and serve as basis to generate *negative topics* – topics that can be disregarded for the current context. We presume that categories of hierarchical taxonomy systems such as Wikipedia categories⁷, YAGO⁸, or Umbel⁹ aggregate entities that belong to similar topics. Under this assumption already eliminated entity candidates can be utilized to successively create the negative context.

This approach of successively creating a negative context requires a high confidence regarding the disambiguation process. If a term is disambiguated incorrectly, potentially “wrong” entities will be added to the negative context. This will bias both the negative and the positive context. Therefore the terms within a context have to be ordered according to the probability that they can be disambiguated correctly. Only disambiguated entities of high confidence will be considered for context creation, as will be shown in the next section.

B. Term Order using a Context Model

For our NEM method, contextual descriptions are used to weight terms within a given context and to derive a confidence value for the subsequent disambiguation. Thus, the natural language terms can be ordered according to their confidence before proceeding with the disambiguation process beginning with the term of highest confidence. This confidence value predicts a probability to disambiguate this term correctly.

For our application, context is determined by natural language text descriptions or text-based metadata originating from metadata of videos. Video metadata cover a broad range of textual metadata types. Such as time-based annotations coming from user-provided tags, or natural language text from authoritative sources, as well as text from automated analysis, such as Video OCR (Optical Character Recognition) or Automated Speech Recognition (ASR). Thus, the processed texts are derived from different sources and of different reliability or confidence.

A context is then composed of several metadata items originating from different sources within a time-based video segment or from authoritative metadata referencing the video as a whole, such as e. g. the title. As shown in [10], the context boundary for video metadata achieving the best analysis results for our purpose is a content-based segment. Therefore in our approach a context is defined to contain all metadata of the same content-based segment.

The terms within a context are ordered according to the following characteristics, which will be explained in more detail:

- text type

⁷<http://en.wikipedia.org/wiki/Wikipedia:FAQ/Categorization>

⁸<http://yago-knowledge.org>

⁹<http://www.umbel.org>

- source reliability
- assigned class cardinality
- source diversity
- number of tokens for a term

The metadata characteristics are described more in detail in the following.

a) *Text Type*: Video metadata is provided in different text types. Descriptive texts and automatically extracted texts are natural language texts. Authoritative metadata can also be provided as typed key-value pairs, or untyped keywords.

b) *Source Reliability*: Metadata provided via automatic extraction methods, as e.g., ASR or Video OCR is considered less reliable than text originating from authoritative sources. Therefore, the source reliability predicts a prospective correctness of the provided textual information.

c) *Source Diversity*: Same natural language terms may be provided by different sources within a video. If a textual information provided by a less reliable source is confirmed by another source, the potential correctness of the information rises. The more sources agree on the same term within the same context the higher is the reliability for this term to be correct.

d) *Class Cardinality*: Conditional Random Field classifier (CRF) as implemented in the Stanford NER tagger¹⁰ identify specific class types of entities found in a text, as e.g. *Organization, Person, Place* etc. Such a classification limits the number of potential entity candidates to the instances of the assigned class. Thereby, the prospective ambiguity decreases and the subsequent disambiguation process has to consider less candidates.

e) *Number of Token*: Labels of semantic entities often consist of more than a single token. The more tokens an extracted natural language term consists of the more specific this term might be considered. Therefore, the prospective ambiguity of a term declines with the increasing number of tokens.

According to these characteristics a confidence value (within the interval [0...1]) for every natural language term within the current context is calculated and all terms are ordered from highest to lowest confidence for subsequent disambiguation. Further details about the applied context model and the scoring algorithm are described in [10].

We now assume to proceed with an ordered list of natural language terms with decreasing confidence according to the defined characteristics. At the beginning of the disambiguation process the negative context is empty. It will grow with every term that is already successfully disambiguated. The process of dynamically building the negative context is described in the next section.

C. Dynamically Building a Negative Context

During the disambiguation process of an ambiguous term all the entity candidates achieve a score denoting how relevant the entity is according to the present context. Usually the entity

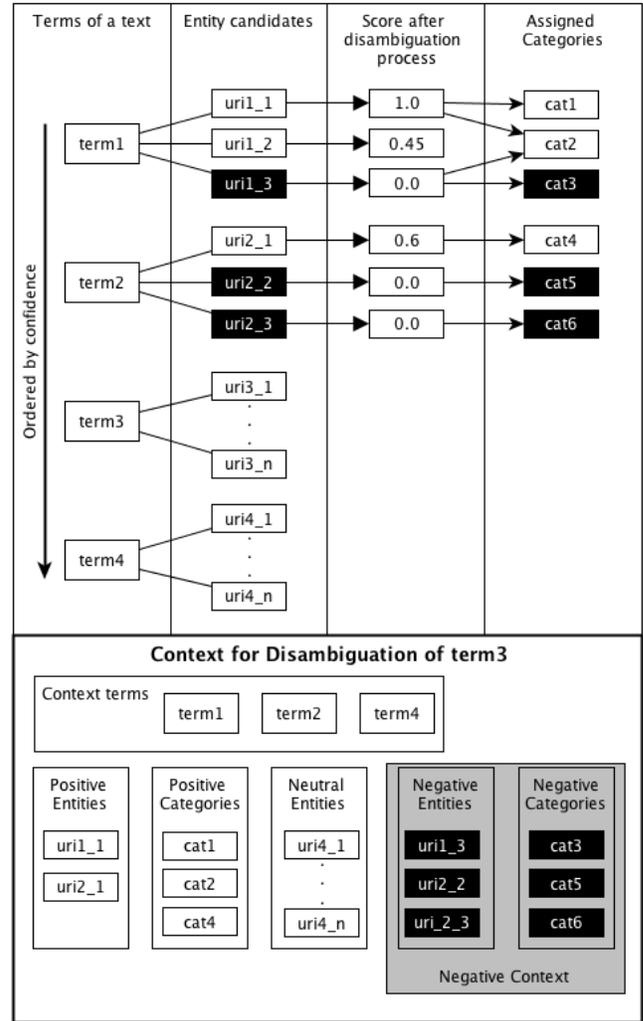


Fig. 2. Creating negative context during disambiguation process

with the highest score “wins” the disambiguation process and is mapped as a *positive entity* to the term. Our approach is based on the presumption that entity candidates, which achieve a disambiguation score of $s_{total} = 0.0$ are considered to be not relevant at all within the current context. Therefore, these discarded entity candidates will furthermore be considered as negative context. Successively these entity candidates will be added to the negative context¹¹ and applied for further disambiguation processes. The negative context can be considered as a set of topics that are not relevant for the present context. Therefore, the negative context also comprises not only individuals, but also more abstract or generic categories. These categories can be simply derived from the `rdf:type` respectively `dc:subject` information assigned to the negative (individual) entities. For every category assigned to a negative entity it is evaluated whether it also assigned to a positive entity, i.e. an entity already confirmed for the current

¹⁰<http://nlp.stanford.edu/software/CRF-NER.shtml>

¹¹subsequently these entities are referred to as *negative entities*

TABLE I
EXTRACTED TERMS OF EXAMPLE SENTENCE AND CALCULATED
CONFIDENCE VALUE FROM CONTEXTUAL DESCRIPTIONS

Term	Confidence
Apple	0.704
operating system	0.612
Jaguar	0.512
Lion	0.512

context. If not, it is added as negative category to the negative context. An overview of the creation of the negative context is depicted in Figure 2.

For our approach DBpedia¹² entities are used as semantic entities. There exist several classification hierarchies that are assigned to DBpedia entities, as e.g. Wikipedia categories, YAGO, or Umbel. Due to its good maintenance and cycle-free hierarchy the YAGO categories are applied for the proposed approach.

Example: Please consider the following sentence: “*Lion and Jaguar are both operating systems from Apple.*” The terms “Lion”, “Jaguar”, “operating system”, and “Apple” are detected as named entities, potential entity candidates are assigned, and a confidence value for the subsequent disambiguation is calculated (cf. Table I). “Apple” is identified as an organization by the CRF tagger (cf. section III-B). For the other detected named entities no classification is assigned. Therefore, “Apple” obtains the highest confidence value and will be disambiguated first. In the following disambiguation process the entity *Apple Inc.*¹³ obtains the highest score compared to other entity candidates for the term “Apple”. The categories for this entity are added to the positive categories of the current context. The entities *Apple (band)*¹⁴, *The Apples (Israeli)*¹⁵ obtained a total score of $s_{total} = 0.0$ during the disambiguation. Therefore they are added to the negative context. None of the assigned categories for these two negative entities are linked to the positive context entity *Apple Inc.*. Therefore, the categories of these discarded entities are added to the negative categories for the current context. The context after disambiguating the term “Apple” is depicted in Table II. With every disambiguated term the negative context and the sets of positive and negative categories grow and can be applied for following disambiguation processes. The influence of the negative context and categories on the disambiguation process is described in the next section.

D. The disambiguation process

We have developed several heuristics using the positive and neutral context to disambiguate terms. All heuristics calculate a score within the interval [0.0...1.0]. The scores are weighted and added up to a total score $s_{total} = [0.0...1.0]$. Heuristics and weights have been determined and evaluated empirically. Further information about the general disambiguation process

¹²<http://dbpedia.org/About>

¹³http://dbpedia.org/resource/Apple_Inc.

¹⁴[http://dbpedia.org/resource/Apple_\(band\)](http://dbpedia.org/resource/Apple_(band))

¹⁵[http://dbpedia.org/resource/The_Apples_\(Israeli\)](http://dbpedia.org/resource/The_Apples_(Israeli))

TABLE II
CONTEXT FOR THE EXAMPLE SENTENCE AFTER DISAMBIGUATING THE
TERM “APPLE”

Positive Entities	<i>Apple Inc.</i> for term <i>Apple</i>
Positive Categories	yago:Company yago:CompaniesEstablishedIn1976 yago:SteveJobs yago:NetworkingHardwareCompanies
Negative Categories	yago:EnglishRockMusicGroups yago:MusicalGroupsEstablishedIn1968 yago:1960sMusicGroups
Neutral Terms	<i>operating system</i> (1 candidate) <i>Lion</i> (88 candidates) <i>Jaguar</i> (58 candidates)

and the scoring methods are described in [11]. In addition, to these positive scorers a new negative scorer has been developed that makes use of the negative context information. It is integrated in the disambiguation process in the following way.

1) *Calculating the Negative Score:* First, for an entity candidate it is checked whether the negative context already contains the entity. If so, the entity candidate is not considered for further scoring and ignored as potential positive entity for the term currently disambiguated.

If the candidate is not part of the negative context, the assigned categories for the entity are retrieved. This set and the set of negative categories in the negative context are examined for an intersection. The negative score assigned to an entity depends on the size of the intersection and on how specific the respective categories in the intersection are. The specificity can be derived from the category’s tree depth within the classification hierarchy. The higher the tree depth the more specific is the category. More general categories usually contain a higher number of entities and the relevance of this category for the considered entities is lower than for more specific categories containing less entities. Thereby, a category can be weighted taking into account its significance for the entity candidate. The weight is calculated from the logarithm of the tree depth proportional to the logarithm of the maximum tree depth within the considered classification system¹⁶. Thereby a weight for the category regarding the entity candidate is achieved within the interval [0.0...1.0]. For the total negative score the weights of all categories in the intersection are added up and divided by the intersection size:

$$s_{negative} = \frac{\sum_{i=1}^n \frac{\log(t_i)}{\log(t_{max})}}{|I|},$$

where I depicts the intersection of the categories in the negative context and the categories assigned to the entity candidate. I contains categories c_i , $c_i \in I$, for $1 \leq i \leq n$. t_i is the tree depth assigned to category c_i , and t_{max} depicts the maximum tree depth determined in the applied category hierarchy.

¹⁶For the YAGO classification system a maximum tree depth of 18 has been calculated.

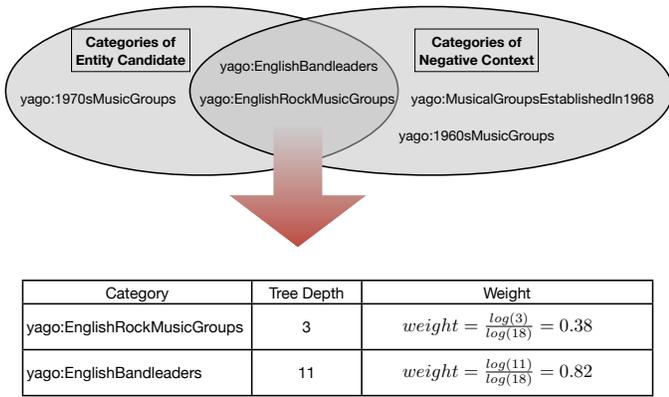


Fig. 3. Intersection of negative and entity candidate categories and influence of tree depth

The approach of achieving a negative score is also depicted in Figure 3.

2) *Calculating the Total Score*: The total score for an entity candidate taking into account the negative context can be calculated in two different ways. Either the *total positive score* is set to zero as soon as the *negative score* is unequal zero. Or the *total positive score* is reduced by the *total negative score* resulting in a final score within the interval $[-1.0...1.0]$. For the latter case the negative score can also be weighted to lower or increase the influence of the negative context on the total score of an entity candidate. Both calculation approaches have been evaluated. The results and discussion of the results are explained in section IV.

3) *Final Decision*: After the scoring process usually the entity candidate that achieved the highest total score is chosen as prospectively correct disambiguation for the respective term. Sometimes it happens that no entity candidate has achieved a positive score, i.e. means that all entity candidates hold a total score $s_{total} = 0.0$. In that case, we have decided to choose the most popular entity candidate according to the amount incoming links within the DBpedia page link graph. In this way, often a boost in terms of recall can be achieved. But, it is important to choose the most popular entity within the set of entity candidates that has *not* achieved a negative score wrt. the negative context.

IV. EVALUATION & DISCUSSION

Improvements of NEM algorithms usually aim to increase recall and precision compared to a manually generated ground truth for given texts. As the proposed approach is mainly developed for improving NEM results on video metadata existing datasets are not suitable as benchmarks for evaluation. We have created a new dataset containing video metadata that is used for the presented evaluation.

Besides slightly increasing recall and/or precision especially for the textual information with lower confidence the significance of the results has been improved. The term significance and its description is introduced in Section IV-B.

TABLE III
EVALUATION RESULTS (R – RECALL, P – PRECISION, AND F_1 -MEASURE) OF THE *conTagger* COMPARED TO *conTagger* USING NEGATIVE CONTEXT

	<i>conTagger</i>			<i>conTagger</i> using negative Context		
	R	P	F_1	R	P	F_1
Authoritative	60.0	54.5	57.0	61.0	55.0	58.0
Tags	71.0	69.5	70.0	71.0	69.5	70.0
ASR	55.0	61.0	58.0	55.0	61.0	58.0
OCR	56.0	24.0	34.0	60.0	26.0	36.5

Evaluation results and a discussion are depicted in sections IV-B and IV-C. The utilized dataset is described in the next section.

A. Evaluation Data Set

Our evaluation dataset consists of video metadata and has already been applied for the evaluation of [10]. This dataset consists of metadata of five videos. The videos are live recordings of TED¹⁷ conference talks covering the topics physics, biology, psychology, sociology, and history science. Overall the dataset consists of 822 metadata items, where an item can be a single key term or a natural language text consisting of up to almost 1000 words. Overall, 2550 entities have been identified¹⁸. The ground truth for the dataset has been annotated manually by five different researchers to provide a reference dataset independent from the opinion of a single person.

B. Results

The evaluation results are shown in Table III. The evaluation compares our proposed approach to the one already presented in [10] – the *conTagger*.

The left three columns show recall, precision, and F_1 -measure for the *conTagger* without negative context. The *conTagger* has been evaluated against current NEM approaches and achieved superior recall and/or precision results, as described in [10]. The right three columns show recall, precision, and F_1 -measure for the extended version of *conTagger* with negative context for the disambiguation. Recall and precision have been calculated for the metadata items of the different sources, ASR, OCR, user tags, and authoritative information, separately and are represented by the respective row in Table III.

As shown, recall and precision are improved by using the negative context especially for metadata items with prospectively lowest confidence, as e. g. OCR metadata, because they have been derived from less reliable sources. Results for tags and items retrieved by ASR nearly remain constant for both approaches with or without negative context. Hereby we are able to show the positive influence of negative context in the disambiguation process. We will take a closer look on the evaluation results in the discussion section.

¹⁷<http://www.ted.com>

¹⁸For details or downloading the dataset and the ground truth please cf. the readme file at <http://bit.ly/15a1YCO>

In addition, to the improvement of recall and precision for metadata item disambiguation, we have detected further evaluation findings: the significance of our disambiguation results has been increased.

After disambiguation the entity candidate with the highest score is chosen as correctly disambiguated entity for the respective natural language term. The distance between the first and the second highest achieved score of the entity candidates for a term can be considered as an indicator for significance and also the reliability of the achieved result. The higher the distance between the first and the second highest score the more reliable the result can be considered. By using negative context this significance has been increased.

The best results have been achieved by reducing the *total positive score* by the computed *total negative score* (using the weight 1.0 equally for both scores). In case the achieved total score is below zero the total score is set to $s_{total} = 0.0$. Thereby the interval for the total score remains within [0.0...1.0].

The averaged significance over all results for our video metadata test set without negative context is 0.27. Taking into account the negative categories the significance amounts to 0.40. Therefore the significance of the analysis results rises by an average of 0.13.

The significance of analysis results is of importance if the decision for chosen entity candidates is determined by applying a threshold on the obtained scores. In this case, a higher significance also results in a higher precision.

C. Discussion of the Results

The application of the negative context and the negative categories aimed at the provision of topics that can be eliminated as relevant topics for the context. Unfortunately, categories do not provide information leading to context relevant topics. The categories derived from the Wikipedia (as YAGO, or the Wikipedia classification) do not supply type comprehensive topic information (as in persons, places, or organizations belonging to a specific subject). The categories are mostly constricted to one specific type – as e.g., the category *yago:EnglishRockMusicGroups* is restricted to bands, but does not imply the topic *music*. Thus, the negative context – consisting of negative entities and negative categories – can not represent negative topics. However, in some cases the disambiguation process has been positively improved by applying the negative context and devaluing prospectively wrong entity candidates. This shows that we are able to construct negative context, that partly represents negative topics. Supposed incorrect entity candidates that also achieved a score within the positive context have been devalued by using the negative context. This is shown by the increase of the recall and the precision.

However, detailed investigation has shown impact of the negative context on terms that are not integrated in a context in general. Common or general terms such as e.g. *history*, *worry*, *audience* are hard to disambiguate. As it is the case for DBpedia, these terms are also often used for band names

or music albums and thus, these entities that belong to a very specific category also end up as candidates for common or general terms. Our evaluation has shown that these types of entities are often linked to the negative context, if the already disambiguated terms contain such entities as candidates. These entity candidates then are devalued and can be disregarded in the decision for the correct disambiguation.

D. Evaluation on independent Datasets

The context model, shortly described in Section III-B, originally has been developed to process video metadata. Most of the contextual characteristics can only be applied for textual information originating from different sources and of different text types. However, the contextual model can also be applied for simple natural language texts to achieve a more reliable disambiguation by ordering the text terms. This likewise enables a disambiguation process beginning with the prospectively most confident term. This in turn enables the successive construction of the negative context. Therefore, we have also tested our approach on a dataset consisting of 10 New York Times articles. This dataset originally has been developed to evaluate DBpedia Spotlight¹⁹ and is described in [8].

Without using a negative context our disambiguation approach achieves a recall of 0.58 and a precision of 0.41 - resulting in a F_1 -measure of 0.48. The approach with negative context achieves a recall of 0.60 and a precision of 0.42 – resulting in a F_1 -measure of 0.495. Here again, the significance of the analysis results is increased by 0.1 – from 0.35 to 0.45 – by using negative context.

The positive impact of a negative context is shown on this dataset, although the approach has not been developed for this type of data.

V. SUMMARY & FUTURE WORK

This paper has presented a new approach for Named Entity Mapping by including negative context. For the generation of a negative context negative entities are retrieved from already discarded entity candidates of disambiguated terms. The order in which the terms of a context have to be analyzed has been determined with the help of contextual descriptions. This order predicts which terms may be disambiguated with a higher confidence than the others. The negative context then successively grows with every analyzed ambiguous term. The negative context consists of the eliminated entity candidates and their assigned categories. The original intention of this approach was to increase NEM performance in terms of recall and precision on given evaluation datasets. An additional finding was an achieved improvement of the significance of the analysis results, while maintaining or even increasing recall and precision. This means that the distance of the scores between the winning and the second place entity within an ordered list of entity candidates increases and thus, the decision becomes more reliable.

¹⁹<https://github.com/dbpedia-spotlight>

Granularity of context for natural language texts influences the specificity of the context. A too fine granular context might provide too little information and a too large context might “soften” the topic. Ongoing work deals with the problem of finding a suitable context granularity for the application of positive and negative context to further improve our results.

Ongoing and future work includes research on building the negative context by using latent topics. Böhm et al. describe an approach on aggregating entities of an RDF graph in subgraphs and thereby building latent topics [12]. The entities of such subgraphs can be added to the negative context, if a sufficient amount of eliminated *negative* entities is part of such a graph. This eliminates specific topics from the relevant topics for the given positive context.

Additionally, other classification systems and heuristics for computing a negative score will be tested. The idea to use negative context for the disambiguation process is very novel but promising. Although recall and precision have not been significantly increased by using the current approach the already achieved results have been consolidated with a higher reliability, which is a solid basis for future work.

REFERENCES

- [1] R. Westenthaler and O. Grisel, “Automated linking data with apache stanbol,” in *Proceedings of the World Wide Web Conference*, 2012.
- [2] M. Bazire and P. Brézillon, “Understanding context before using it,” in *Proceedings of the 5th international conference on Modeling and Using Context*, ser. CONTEXT’05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 29–40. [Online]. Available: http://dx.doi.org/10.1007/11508373_3
- [3] T. van der Wouden, *Negative Contexts - Collocation, Polarity and Multiple Negation*. Routledge Chapman & Hall, 1997.
- [4] J. Domínguez-Borràs, M. Garcia-Garcia, and C. Escera, “Negative emotional context enhances auditory novelty processing: behavioural and electrophysiological evidence,” *European Journal of Neuroscience*, vol. 28, no. 6, pp. 1199–1206, 2008. [Online]. Available: <http://dx.doi.org/10.1111/j.1460-9568.2008.06411.x>
- [5] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Found. Trends Inf. Retr.*, vol. 2, no. 1-2, pp. 1–135, Jan. 2008. [Online]. Available: <http://dx.doi.org/10.1561/1500000011>
- [6] P. McCrae, “A computational model for the influence of cross-modal context upon syntactic parsing,” Ph.D. dissertation, Universität Hamburg, Von-Melle-Park 3, 20146 Hamburg, 2010. [Online]. Available: <http://ediss.sub.uni-hamburg.de/volltexte/2010/4800>
- [7] V. Bryl, C. Giuliano, L. Serafini, and K. Tymoshenko, “Supporting natural language processing with background knowledge: coreference resolution case,” in *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*, ser. ISWC’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 80–95. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1940281.1940288>
- [8] P. N. Mendes, M. Jacob, A. Garcia-Silva, and C. Bizer, “Dbpedia spotlight: Shedding light on the web of documents,” in *Proc. of 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria*, 2011.
- [9] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.
- [10] N. Steinmetz and H. Sack, “Semantic multimedia information retrieval based on contextual descriptions,” in *Proceedings of the 10th Extended Semantic Web Conference*, ser. LNCS. Springer, May 2013, Inproceedings, to appear.
- [11] N. Ludwig and H. Sack, “Named entity recognition for user-generated tags,” in *Proceedings of the 2011 22nd International Workshop on Database and Expert Systems Applications*, ser. DEXA ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 177–181. [Online]. Available: <http://dx.doi.org/10.1109/DEXA.2011.56>
- [12] C. Böhm, G. Kasneci, and F. Naumann, “Latent topics in graph-structured data,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, ser. CIKM ’12. New York, NY, USA: ACM, 2012, pp. 2663–2666. [Online]. Available: <http://doi.acm.org/10.1145/2396761.2398718>