Keyword Query Expansion on Linked Data Using Linguistic and Semantic Features

Saeedeh Shekarpour*, Konrad Höffner*, Jens Lehmann* and Sören Auer*
University of Leipzig, Department of Computer Science
Augustusplatz 10, 04109 Leipzig
{lastname}@informatik.uni-leipzig.de

***Abstract*—Effective search in structured information based on textual user input is of high importance in thousands of applications. Query expansion methods augment the original query of a user with alternative query elements with similar meaning to increase the chance of retrieving appropriate resources. In this work, we introduce a number of new query expansion features based on semantic and linguistic inferencing over Linked Open Data. We evaluate the effectiveness of each feature individually as well as their combinations employing several machine learning approaches. The evaluation is carried out on a training dataset extracted from the QALD question answering benchmark. Furthermore, we propose an optimized linear combination of linguistic and lightweight semantic features in order to predict the usefulness of each expansion candidate. Our experimental study shows a considerable improvement in precision and recall over baseline approaches.**

I. Introduction

With the growth of the Linked Open Data cloud, a vast amount of structured information was made publicly available. Querying that huge amount of data in an intuitive way is challenging. SPARQL, the standard query language of the semantic web, requires exact knowledge of the vocabulary and is not accessible by laypersons. Several tools and algorithms have been developed that make use of semantic technologies and background knowledge [10], such as *TBSL* [17], *SINA* [16] and Wolfram|Alpha[1]. Those tools suffer from a mismatch between query formulation and the underlying knowledge base structure that is known as the *vocabulary problem* [8]. For instance, using DBpedia as knowledge base, the query "Who is married to Barack Obama?" could fail, because the desired property in DBpedia is labeled "spouse" and there is no property labeled "married to".

Automatic query expansion (AQE) is a tried and tested method in web search for tackling the vocabulary problem by adding related words to the search query and thus increase the likelihood that appropriate documents are contained in the result. The query is expanded with features such as synonyms, e.g. "spouse" and "married to" in the example above, or *hyponym-hypernym* relations, e.g. "red" and "color". We investigate, which methods semantic search engines can use to overcome the vocabulary problem and how effective AQE with the traditional linguistic features is in this regard. Semantic search engines can use the graph structure of RDF and follow interlinks between datasets. We employ this to generate additional expansion features such as the labels of sub- and superclasses of resources. The underlying research question is whether interlinked data and vocabularies provide features which can be taken into account for query expansion and how effective those new semantic query expansion features are in comparison to traditional linguistic ones.

We do this by using machine learning methods to generate a linear combination of linguistic and semantic query expansion features with the aim of maximizing the F_1-score and efficiency on different benchmark datasets. Our results allow developers of new search engines to integrate AQE with good results without spending much time on its design. This is important, since query expansion is usually not considered in isolation, but rather as one step in a pipeline for question answering or keyword search systems.

Our core contributions are as follows:

- Definition of several semantic features for query expansion.
- Creation of benchmark test sets for query expansion.
- Combining semantic and linguistic features in a linear classifier.
- An analysis of the effect of each feature on the classifier as well as other benefits and drawbacks of employing semantic features for query expansion.

The paper is structured as follows: In section II, the overall approach is described, in particular the definition of features and the construction of the linear classifier. Section III provides the experiment on the QALD-1, QALD-2 and QALD-3 test sets and presents the results we obtained. In the related work in section IV, we discuss in which settings AQE is used. Finally, we conclude and give pointers to future work.

II. Approach

In document retrieval, many query expansion techniques are based on information contained in the top-ranked retrieved documents in response to the original user query, e.g. [6]. Similarly, our approach is based on performing an initial retrieval of resources according to the original keyword query. Thereafter, further resources are derived by leveraging the initially retrieved ones. Overall, the proposed process depicted in the Figure 1 is divided into three main steps. In the first step, all words closely related to the original keyword are extracted based on two types of features – linguistic and semantic. In the second step, various introduced linguistic and semantic

[1] http://www.wolframalpha.com

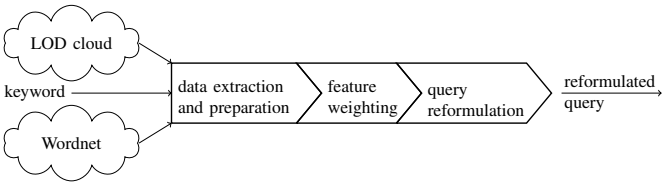

Fig. 1. AQE Pipeline.

features are weighted using learning approaches. In the third step, we assign a relevance score to the set of the related words. Using this score we prune the related word set to achieve a balance between *precision* and *recall*.

A. Extracting and Preprocessing of Data using Semantic and Linguistic Features

For the given input keyword k, we define the set of all words related to the keyword k as $X_k = \{x_1, x_2, ..., x_n\}$. The set X_k is defined as the union of the two sets LE_k and SE_k. LE_k (resp. SE_k) is constructed as the collection of all words obtained through linguistic features (resp. semantic). Linguistic features extracted from WordNet are:

- *synonyms:* words having a similar meanings to the input keyword k.
- *hyponyms:* words representing a specialization of the input keyword k.
- *hypernyms:* words representing a generalization of the input keyword k.

The set SE comprises all words semantically derived from the input keyword k using Linked Data. To form this set, we match the input keyword k against the `rdfs:label` property of all resources available as Linked Open Data[2]. It returns the set $AP_k = \{r_1, r_2, ..., r_n\}$ as $AP_k \subset (C \cup I \cup P)$ where C, I and P are the sets of classes, instances and properties contained in the knowledge base respectively, whose labels contain k as a sub-string or are equivalent to k. For each $r_i \in AP_k$, we derive the resources semantically related to r_i by employing a number of semantic features. These semantic features are defined as the following semantic relations:

- *sameAs:* deriving resources having the same identity as the input resource using `owl:sameAs`.
- *seeAlso:* deriving resources that provide more information about the input resource using `rdfs:seeAlso`.
- *class/property equivalence:* deriving classes or properties providing related descriptions for the input resource using `owl:equivalentClass` and `owl:equivalentProperty`.
- *superclass/-property:* deriving all super classes/properties of the input resource by following the `rdfs:subClassOf` or `rdfs:subPropertyOf` property paths originating from the input resource.
- *subclass/-property:* deriving all sub resources of the input resource r_i by following the `rdfs:subClassOf` or `rdfs:subPropertyOf` property paths ending with the input resource.
- *broader concepts:* deriving broader concepts related to the input resource r_i using the SKOS vocabulary properties `skos:broader` and `skos:broadMatch`.
- *narrower concepts:* deriving narrower concepts related to the input resource r_i using `skos:narrower` and `skos:narrowMatch`.
- *related concepts:* deriving related concepts to the input resource r_i using `skos:closeMatch`, `skos:mappingRelation` and `skos:exactMatch`.

Note that on a given r_i only those semantic features are applicable which are consistent with its associated type. For example, sameAs only. For instance, super/sub class/property relations are solely applicable to resources of type `class` or `property`.

For each $r_i \in AP_k$, we derive all the related resources employing the above semantic features. Then, for each derived resource r', we add all the English labels of that resource to the the set SE_k. Therefore, SE_k contains the labels of all semantically derived resources. As mentioned before, the set of all related words of the input keyword k is defined as $X_k = LE_k \cup SE_k$. After extracting the set X_k of related words, we run the following preprocessing methods for each $x_i \in X_k$:

1) *Tokenization:* extraction of individual words, ignoring punctuation and case.
2) *Stop word removal:* removal of common words such as articles and prepositions.
3) *Word lemmatisation:* determining the lemma of the word.

Vector space of a word: A single word $x_i \in X_k$ may be derived via different features. For example, as can be observed in Figure 2, the word "motion picture" and "film" is derived by *synonym*, *sameAs* and *equivalent* relations. Thus, for each derived word x_i, we define a vector space representing the derived features resulting in including that word. Suppose that totally we have n linguistic and semantic features. Each $x_i \in X_k$ is associated with a vector of size n as $V_{x_i} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$. Each α_i represents the presence or absence of the word x_i in the list of the words derived via the feature f_i. For instance, if we assume that f_1 is dedicated to the *synonym* feature, the value 1 for α_1 in the V_{x_i} means that x_i is included in the list of the words derived by the *synonym* feature. There are features and those features generate a set of expansion words.

B. Feature Selection and Feature Weighting

In order to distinguish how effective each feature is and to remove ineffective features, we employ a weighting schema ws for computing the weights of the features as $ws : f_i \in F \rightarrow w_i$. Note that F is the set of all features taken into account. There are numerous feature weighting methods to

[2]via http://lod.openlinksw.com/sparql

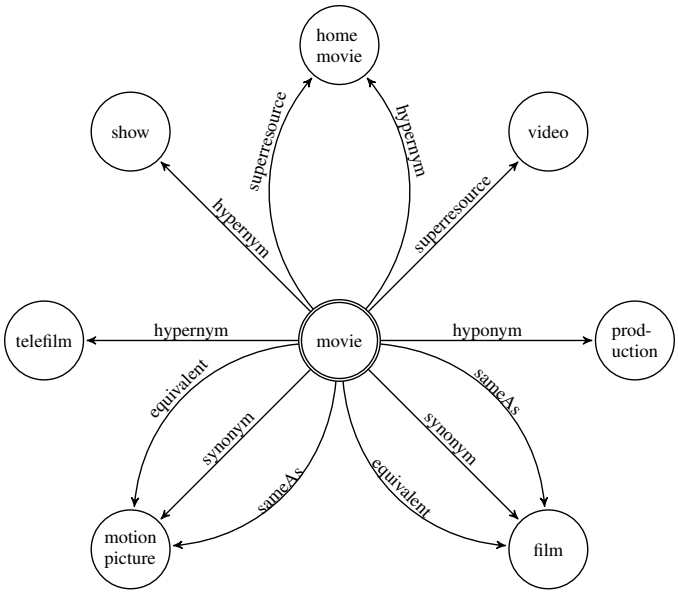

Fig. 2. Exemplary expansion graph of the word *movie* using semantic features.

assign weight to features like information gain [7], weights from a linear classifier [15], odds ratio, etc. Herein, we consider two well-known weighting schemas.

1) Information Gain (IG): Information gain is often used (see section IV) to decide which of the features are the most relevant. We define the information gain (IG) of a feature as:

$$\mathrm{IG}(f_i) = \sum_{\substack{c\in\{+,-\} \\ f_i\in\{present,absent\}}} \Pr(c, f_i) \ln \frac{\Pr(c, f_i)}{\Pr(f_i)\Pr(c)}$$

In our approach, we used the ID3 decision tree algorithm with information gain.

2) Feature weights from linear classifiers: Linear classifiers, such as for example SVMs, calculate predictions by associating the weight w_i to the feature f_i. Features whose w_i is close to 0 have a small influence on the predictions. Therefore, we can assume that they are not very important for query expansion.

C. Setting the Classifier Threshold

As a last step, we set the threshold for the classifiers above. To do this, we compute the relevance score value $\mathrm{score}(x_i)$ for each word $x_i \in X_k$. Naturally, this is done by combining the feature vector $V_{x_i} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ and the feature weight vector $W = [w_1, w_2, \ldots, w_n]$ as follows:

$$\mathrm{score}(x_i) = \sum_{i=1:n} \alpha_i w_i$$

.

We can then compute a subset Y_k of X_k by selecting all words, which are above a threshold θ:

$$Y_k = \{y | y \in X_k \wedge \mathrm{score}(y) \geq \theta\}$$

.

This reduces the set of query expansion candidates X_k to a set Y_k containing only the words in X_k above threshold θ. Since we use a linear weighted combination of features, words which exhibit one or more highly relevant features are more likely to be included in the query expansion set Y_k. Thus, the threshold can be used to control the trade-off between precision and recall. A high threshold means higher precision, but lower recall whereas a low threshold improves recall at the cost of precision.

To sum up, Y_k is the output of the AQE system and provides words semantically related to the input keyword k. Retrieval of resources from the underlying knowledge base is based on the match of a given keyword with the `rdfs:label` of resources. Thus, this expansion increases the likelihood of recognizing more relevant resources. Because in addition to resources matching their `rdfs:label` to the input keyword k, we take into account resources having match of their `rdfs:label` with the words contained in Y. Subsequently, this causes an increase in *recall*. On the contrary, it may result in loss of *precision* by including resources which may be irrelevant. A severe loss in *precision* significantly hurts retrieval performance. Therefore, we investigate a moderate tradeoff between speed and accuracy, although the final result highly depends on requirements of the search service(precision is more important or recall). Herein, the set Y includes all $x_i \in X$ with a high relevance likelihood and excludes those with a low likelihood.

III. Experiment and Result

A. Experimental Setup

The goal of our evaluation is to determine (1) How effective linguistic as well as semantic query expansion features perform and (2) How well a linear weighted combination of features performs. To the best of our knowledge, no benchmark has been created so far for query expansion tasks over Linked Data. Thus, we created one benchmark dataset. This dataset called QALD benchmark contains 37 keyword queries obtained from the *QALD-1* and *QALD-2* benchmarks[3]. The *QALD-1* and *QALD-2* benchmarks are essentially tailored towards comparing question answering systems based on natural language queries. We extracted all those keywords contained in the natural language queries requiring expansion for matching to the target knowledge base resource.

An example are the keywords "wife" and "husband" which should be matched to `dbpedia-owl:spouse`. Note that in the following all experiments are done on the dataset except the last experiment.

B. Results

Generally, when applying expansion methods, there is a risk of yielding a large set of irrelevant words, which can have a negative impact on further processing steps. For this reason, we were interested in measuring the effectiveness of all

[3]http://www.sc.cit-ec.uni-bielefeld.de/qald-n for $n = 1, 2$.

Features	derived words	matches
synonym	503	23
hyponym	2703	10
hypernym	657	14
sameAs	2332	12
seeAlso	49	2
equivalent	2	0
super class/property	267	4
sub class/property	2166	4
broader concepts	0	0
narrower concepts	0	0
related concepts	0	0

TABLE I: Number of the words derived by each feature and their associated matches.

Feature	GR	SVM	IG
synonym	0.920	0.400	0.300
hyponym	0.400	0.810	0.490
hypernym	1	0.400	0.670
sameAs	0.400	0.800	0.490
seeAlso	0.400	1.360	0.300
equivalent	0.300	0.490	0.300
super class/property	0.450	1.450	0.700
sub class/property	1.500	0.670	1.120

TABLE II: Computed weights of the features using the schemas *Gain Ratio (GR)*, *Support Vector Machines (SVM)* and *Information Gain (IG)*.

linguistic as well as semantic features individually in order to decide which of those are effective in query expansion. Table I shows the statistics over the number of derived words and the number of matches per feature. Interestingly, *sameAs* has even more matches than *synonym*, but also leads to a higher number of derived words. The *hyponym* and *sub class/property* return a huge number of derived words while the number of matches are very low. The features *hypernym* and *super class/property* in comparison to the rest of the features result in a considerable number of matches. The features *broader concepts*, *narrower concepts* and *related concepts* provide a very small amount of derived words and zero number of matches. Thus, we exclude the *skos* features in the later experiments.

In continuation, we investigate the accuracy of each individual feature. Thus, we employ an `svm` classifier and individually make an evaluation over the accuracy of each feature. This experiment was done on the dataset with 10 fold cross validation. Table IV presents the results of this study. In this table, the *precision*, *recall* and *F-Measure* for the positive (+), negative(-) and all (total) examples are shown. In addition, six separate evaluations are carried out over a subset of features which are semantically close to each other e.g. *hyponym+sub class/property*.

In the following, we only mention the most prominent observations of this study. The features *hyponym*, *super class/property* and *sub class/property* have the highest value for *F-Measure*. The *precision* of *sameAs* is the same as for *synonym*. The feature *equivalent* has a high *precision* although the *recall* is very low. The *precision* of the *sub class/property* and *hyponym* is high for negative examples. At last, the combined features always behave better than the individual features.

The second goal of our experimental study is how well a linear weighted combination of features can predict the relevant words? To do that, firstly we employed two weighting schemas as described in the approach, i.e. information gain (IG) and the weighting of the linear classifier `svm`. Secondly, these computed weights are used for reformulating the input keyword.

Table II shows the weights computed by *SVM* schemas. The feature *super class/property* is ranked as the highest distinguishing feature. The computed value of *Hyponym* is relatively high but this feature has a negative influence on all examples. Interestingly, in *SVM* schema *sameAs* and *seeAlso* as well as *synonym* are acquired the equal values. This may result in that *sameAs* and *seeAlso* can be a comparable alternative for *synonym*. Furthermore, *subproperty* and *subclass* are excluded in this schema.

Thereafter, we scored all the derived words according to the beforehand computed weights. We set up two different settings. In each setting, respectively, only linguistic features and only semantic features are taken into account. A separate evaluation is carried out for each setting with respect to the computed weights *SVM*.

Table III shows the results of this study. Interestingly, the setting with only semantic features result in an accuracy at least as high as the setting with only linguistic features. This observation is an important finding of this paper that semantic features appear to be competitive with linguistic features.

Features	Weighting	P	Recall	F-Score
Linguistic	SVM	0.730	0.650	0.620
Semantic	SVM	0.680	0.630	0.600
Linguistic	Decision Tree / IG	0.588	0.579	0.568
Semantic	Decision Tree / IG	0.755	0.684	0.661

TABLE III: Accuracy results of predicting the relevant derived words.

Features	P	Recall	F-Score	
synonym	0.440	0.680	0.540	−
	0.330	0.150	0.210	+
	0.390	0.420	0.370	total
hyponym	0.875	0.368	0.519	−
	0.600	0.947	0.735	+
	0.737	0.658	0.627	total
hypernym	0.545	0.947	0.692	−
	0.800	0.211	0.333	+
	0.673	0.579	0.513	total
sameAs	0.524	0.579	0.550	−
	0.529	0.474	0.500	+
	0.527	0.526	0.525	total
seeAlso	0.471	0.842	0.604	−
	0.250	0.053	0.087	+
	0.360	0.447	0.345	total
equivalent	0.480	0.890	0.630	−
	0.330	0.053	0.091	+
	0.410	0.470	0.360	total
super class/property	0.594	1	0.745	−
	1	0.316	0.480	+
	0.797	0.658	0.613	total
sub class/property	0.480	0.890	0.630	−
	0.330	0.050	0.090	+
	0.520	0.410	0.470	total
sameAs, seeAlso, equivalent	0.500	0.579	0.530	−
	0.500	0.420	0.450	+
	0.500	0.500	0.490	total
synonym, sameAs, seeAlso, equivalent	0.470	0.579	0.520	−
	0.460	0.360	0.410	+
	0.470	0.470	0.460	total
hyponym, subresource	0.875	0.368	0.519	−
	0.600	0.947	0.735	+
	0.737	0.658	0.627	total
hypernym, superresource	0.594	1	0.745	−
	1	0.316	0.480	+
	0.797	0.658	0.613	total

TABLE IV: Separate evaluations of the precision, recall and f-score of each individual feature.

IV. Related Work

Automatic Query Expansion (AQE) is a vibrant research field and there are many approaches that differ in the choice and combination of techniques.

A. Design choices for query expansion

In the following, we describe the most important choices of data sources, candidate feature extraction methods and representations, feature selection methods and expanded query representations (cf. [5] for a detailed survey).

a) Data sources: are organized collections of words or documents. The choice of the data source is crucial because it influences the size of the vocabulary as well as the available relations; and thus possible expansion features. Furthermore, a data source is often restricted to a certain domain and thus constitutes a certain context for the search terms. For example, corpora extracted from newspapers yield good results when expanding search queries related to news but are generally less suited for scientific topics. Thus a search query with the keyword "fields" intended for electromagnetic fields could be expanded to "football fields" and yield even worse results then the original query.

Popular choices for data sources are text corpora, WordNet synsets, hyponyms and hypernyms, anchor texts, search engine query logs or top ranked documents. Our approach uses both WordNet as a source for synonyms, hyponyms and hypernyms as well as the LOD cloud to obtain labels of related classes or properties, such as equivalent, sub- and super-resources (cf. section II). WordNet is used frequently but it suffers from two main problems [5]:

1) There is a lack of proper nouns which we tackle by using the LOD cloud including DBpedia which contains mainly instances.
2) The large amount of ambiguous terms leads to a disambiguation problem. However, this does not manifest itself in the relatively small models of the benchmarks we used (cf. section III). Disambiguation is not the focus of this work but may need to be addressed separately when our approach is used in larger domains.

b) Feature selection: [13] consists of two parts: (1) feature weighting assigns a scoring to each feature and (2) the feature selection criterion determines which of the features to retain based on the weights. Some common feature selection methods are *mutual information*, *information gain*, *divergence from randomness* and *relevance models*. A framework for feature weighting and selection methods is presented in [13]. The authors compare different feature ranking schemes and (although not the primary focus of the article) show that SVMs achieve the highest F_1-score of the examined methods. We use information gain and SVMs separately and compare the results. To learn the feature rankings we use the data mining software *Weka* [12].

c) Expanded query representation: can take the form of a Boolean or structured query, vectors of concept types or unweighted terms and many others. Because our query is a set of keywords, our extended query is an extended set of keywords and thus consists of unweighted terms.

B. Semantic Search and Question Answering Engines

While AQE is prevalent in traditional web search engines, the semantic search engines we examine in the following either do not address the vocabulary problem or tackle it in a different way.

Table V shows how the participants of the QALD/ILD 2012 workshop and selected other approaches tackle the vocabulary problem. Interestingly, two of the three considered approaches did not use any kind of query expansion, relying instead only on exact string matching between the query keyword and the label of the associated resource. *Alexandria* [18] uses *Freebase* to include synonyms and different surface forms.

MHE[4] combines query expansion and entity recognition by using textual references to a concept and extracting Wikipedia anchor texts of links. For example, when looking at the following link:

```
<a href="http://en.wikipedia.org/
wiki/United_Nations">UN</a>
```

Here the word "UN" is mapped to the concept *United_Nations*. This approach takes advantage of a large amount of hand-made mappings that emerge as a byproduct. However, this approach is only viable for Wikipedia-DBpedia or other text corpora whose links are mapped to resources.

Engine	Method
TBSL [17]	WordNet synonyms and BOA pattern library [9]
PowerAqua [14]	WordNet synonyms and hypernyms, `owl:sameAs`
Eager [11]	resources of the same type using Wikipedia categories
Alexandria [18]	alternative names (synonyms and different surface forms) from Freebase [3]
SemSeK [1]	no AQE
QAKiS [4]	no AQE
MHE	Wikipedia anchor texts from links pointing to the concept

TABLE V: Prevalence of AQE in RDF based search or question answering engines.

Eager [11] expands a set of resources with resources of the same type using DBpedia and Wikipedia categories (instead of linguistic and semantic features in our case). Eager extracts implicit category and synonym information from abstracts and redirect information, determines additional categories from the DBpedia category hierarchy and then extracts additional resources which have the same categories in common.

PowerAqua [14] is an ontology-based system that answers natural language queries and uses WordNet synonyms and hypernyms as well as resources related with the `owl:sameAs` property.

[4]http://ups.savba.sk/~marek

A prerequisite of feature-based Query Expansion is, that the data about the features used, i.e. the pairs contained in the relations, is available. Relation equivalency (`owl:equivalentProperty`) links in particular are often not, or not completely, defined for a knowledge base. There is however an approach for mining equivalent relations from Linked Data, that relies on three measures of equivalency: *triple overlap*, *subject agreement* and *cardinality ratio*. [19]

An approach similar to ours is [2], however it relies on supervised learning and uses only semantic expansion features instead of a combination of both semantic and linguistic ones.

V. Conclusions

Semantic search is one of the most important applications for demonstrating the value of the semantic web to real users. In the last years, a number of approaches for semantic search have been introduced. However, other than in traditional search, the effect of query expansion has not yet been studied. With semantically structured knowledge, we can also employ additional semantic query expansion features. In this work, we performed a comprehensive study of semantic query expansion. We compared the effectiveness of linguistic and semantic query expansion as well as their combination. Based on a query expansion benchmark we created, our results suggest that semantic features are at least as effective as linguistic ones and the intelligent combination of both brings a boost in precision and recall.

References

[1] N. Aggarwal and P. Buitelaar. A system description of natural language query over dbpedia. In C. Unger, P. Cimiano, V. Lopez, E. Motta, P. Buitelaar, and R. Cyganiak, editors, *Proceedings of Interacting with Linked Data (ILD 2012), workshop co-located with the 9th Extended Semantic Web Conference, May 28, 2012, Heraklion, Greece*, pages 97–100, 2012.

[2] I. Augenstein, A. L. Gentile, B. Norton, Z. Zhang, and F. Ciravegna. Mapping keywords to linked data resources for automatic query expansion. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, Lecture Notes in Computer Science. Springer, 2013.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[4] E. Cabrio, A. P. Aprosio, J. Cojan, B. Magnini, F. Gandon, and A. Lavelli. Qakis @ qald-2. In C. Unger, P. Cimiano, V. Lopez, E. Motta, P. Buitelaar, and R. Cyganiak, editors, *Proceedings of Interacting with Linked Data (ILD 2012), workshop co-located with the 9th Extended Semantic Web Conference, May 28, 2012, Heraklion, Greece*, pages 88–96, 2012.

[5] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, jan 2012.

[6] K. Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *CIKM*. ACM, 2009.

[7] H. Deng, G. C. Runger, and E. Tuv. Bias of importance measures for multi-valued attributes and solutions. In *ICANN (2)*, volume 6792, pages 293–300. Springer, 2011.

[8] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *COMMUNICATIONS OF THE ACM*, 30(11):964–971, 1987.

[9] D. Gerber and A.-C. Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*, 2011.

[10] R. Guha, R. McCool, and E. Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 700–709, New York, NY, USA, 2003. ACM.

[11] O. Gunes, C. Schallhart, T. Furche, J. Lehmann, and A.-C. N. Ngomo. Eager: extending automatically gazetteers for entity recognition. In *Proceedings of the 3rd Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources and their Applications to NLP*, 2012.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, nov 2009.

[13] S. Li, R. Xia, C. Zong, and C.-R. Huang. A framework of feature selection methods for text categorization. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 692–700, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[14] V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: supporting users in querying and exploring the semantic web content. In *Semantik Web Journal*, page 17. IOS Press, March 2011.

[15] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling. Feature selection using linear classifier weights: interaction with classification models. In *In Proceedings of the 27th Annual International ACM SIGIR Conference (SIGIR2004*. ACM, 2004.

[16] S. Shekarpour, S. Auer, and A.-C. Ngonga Ngomo. Question answering on interlinked data. In *Proceedings of WWW*, 2013.

[17] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 639–648, 2012.

[18] M. Wendt, M. Gerlach, and H. Duwiger. Linguistic modeling of linked open data for question answering. In C. Unger, P. Cimiano, V. Lopez, E. Motta, P. Buitelaar, and R. Cyganiak, editors, *Proceedings of Interacting with Linked Data (ILD 2012), workshop co-located with the 9th Extended Semantic Web Conference, May 28, 2012, Heraklion, Greece*, pages 75–87, 2012.

[19] ziqi zhang, A. L. Gentile, I. Augenstein, E. Blomqvist, and F. Ciravegna. Mining equivalent relations from linked data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sofia, Bulgaria, august 2013.