Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks

Francesco Osborne(✉) and Enrico Motta

Knowledge Media Institute, The Open University, Milton Keynes MK7 6AA, UK
{francesco.osborne,enrico.motta}@open.ac.uk

Abstract. The amount of scholarly data available on the web is steadily increasing, enabling different types of analytics which can provide important insights into the research activity. In order to make sense of and explore this large-scale body of knowledge we need an accurate, comprehensive and up-to-date ontology of research topics. Unfortunately, human crafted classifications do not satisfy these criteria, as they evolve too slowly and tend to be too coarse-grained. Current automated methods for generating ontologies of research areas also present a number of limitations, such as: i) they do not consider the rich amount of indirect statistical and semantic relationships, which can help to understand the relation between two topics – e.g., the fact that two research areas are associated with a similar set of venues or technologies; ii) they do not distinguish between different kinds of hierarchical relationships; and iii) they are not able to handle effectively ambiguous topics characterized by a noisy set of relationships. In this paper we present Klink-2, a novel approach which improves on our earlier work on automatic generation of semantic topic networks and addresses the aforementioned limitations by taking advantage of a variety of knowledge sources available on the web. In particular, Klink-2 analyses networks of research entities (including papers, authors, venues, and technologies) to infer three kinds of semantic relationships between topics. It also identifies ambiguous keywords (e.g., "ontology") and separates them into the appropriate distinct topics – e.g., "ontology/philosophy" vs. "ontology/semantic web". Our experimental evaluation shows that the ability of Klink-2 to integrate a high number of data sources and to generate topics with accurate contextual meaning yields significant improvements over other algorithms in terms of both precision and recall.

Keywords: Scholarly data · Ontology learning · Bibliographic data · Scholarly ontologies · Data mining

1 Introduction

The amount of scholarly data available on the web is steadily increasing, enabling different types of analytics which can provide important insights into the research activity. Increasingly, Semantic Web standards are being used to represent this complex data and, as a result, we have seen the emergence of a number of bibliographic

M. Arenas et al. (Eds.): ISWC 2015, Part I, LNCS 9366, pp. 408–424, 2015.
DOI: 10.1007/978-3-319-25007-6_24

repositories in the Linked Data Cloud [1, 2, 3] and a variety of ontologies to describe scholarly data, including SWRC[1], BIBO[2], BiDO[3], AKT[4] and FABIO[5]. The semantic enhancement of scholarly articles, known as *semantic publishing* [4], is also becoming an important topic, attracting the interest of major publishers and leading to the formation of new communities (e.g., FORCE11[6]), workshops (e.g., Linked Science at ISWC, Sepublica at ESWC, SAVE-SD at WWW), and challenges (e.g., the ESWC Semantic Publishing Challenge[7]).

Indeed, today's scientific knowledge is so vast that scientists necessarily tend to specialize in relatively narrow fields, thus potentially missing important links across different fields and/or ending up reinventing solutions already available in other domains. However, there is growing consensus that semantic technologies can help to overcome this problem by improving our ability to discover, query, explore, annotate and visualize research information on the web [4, 5, 6, 7, 8, 9, 10]. Nonetheless, we still face some important technical challenges before this vision can be realized. These crucially include the problem of identifying and modelling the various relationships that exist between components of the research environment. While this task is relatively easy when describing the relationships between real world entities, such as authors and organizations, it becomes much harder when taking in consideration abstract concepts, such as the notion of *research topic*. For example, while it is easy to retrieve all the co-authors of Enrico Motta, it is much more difficult to identify all the papers of Enrico Motta which are relevant to research on the Semantic Web or one of its sub-areas. For this reason many popular systems for the exploration of research data, such as Google Scholar[8], Microsoft Academic Search[9] and Scopus[10], sidestep the challenge of identifying research topics and linking them to other relevant research entities, and simply use keywords as proxy. Unfortunately, this purely syntactic solution is unsatisfactory, as it fails i) to distinguish research topics from other keywords which can be used to annotate papers; ii) to deal with situations where multiple labels exist for the same research area; iii) to deal with the fact that a keyword may denote different topics depending on the context, and iv) to model and take advantage of the semantic relationships that hold between research areas, treating them instead as lists of unstructured keywords.

The traditional way to address the problem of identifying and structuring research topics has been to adopt human-crafted taxonomies, such as the ACM Computing Classification System[11]. Unfortunately, as we discussed in [11], this solution also presents a number of problems. First, building a large taxonomy of research areas requires a large number of experts and is an expensive and lengthy process.

[1] http://ontoware.org/swrc/

[2] http://bibliontology.com.

[3] http://purl.org/spar/bido

[4] http://www.aktors.org/publications/ontology

[5] http://purl.org/spar/fabio

[6] https://www.force11.org

[7] https://github.com/ceurws/lod/wiki/SemPub2015

[8] https://scholar.google.com

[9] http://academic.research.microsoft.com/

[10] http://www.scopus.com/

[11] http://www.acm.org/about/class/2012

For example, the 2012 version of ACM taxonomy was finalized fourteen years after the previous version. Hence, by the time these taxonomies are released they tend to be already obsolete, especially in fields such as Computer Science, where the most interesting topics are the newly emerging ones. Moreover, these taxonomies are very coarse-grained and usually represent wide categories of approaches, rather then the fine-grained topics addressed by researchers. For example, in the ACM Classification, the Semantic Web area is characterized as "Semantic web description languages" and has only two sub-areas: "OWL" and "RDF". Finally, these taxonomies are ambiguous, since the semantics of their links is not specified.

For these reasons, it is our view that building large-scale and timely taxonomies of research topics is a task that needs to be tackled through automatic methods and in 2012 we developed Klink [11], an algorithm which takes as input large amounts of scholarly metadata and automatically generates an OWL ontology containing all the research areas mined from the input data and their semantic relationships. This approach was demonstrated to work very well in comparison with the state of art and the ontology produced by Klink has been used to provide a comprehensive semantic topic network for Rexplore [5], a novel system which integrates semantic technologies, statistical analysis and visual analytics to provide effective support for making sense of scholarly data. In particular, the ontology generated by Klink enhances semantically a variety of data mining and information extraction techniques, and improves search and visual analytics. A variation of Klink was also used in the field of recommender systems to improve significantly the performance of a state of the art content-based recommender [12].

However, both Klink and similar solutions – e.g., [8, 13, 14], suffer from a number of limitations. First, they only consider the graph of co-occurrences between keywords [11] and/or direct semantic relationships [12], thus ignoring relevant indirect statistical and semantic relationships – e.g., the situation where two topics are related to the same conferences or associated to the same standards, knowledge which can improve the robustness and the performance of a solution, especially in the presence of noisy data. Moreover, they fail to deal with keywords which can denote different topics depending on the context in which they are used – e.g., "java" can be a programming language, but also an Indonesian island.

To address these problems we have developed Klink-2, an evolution of the Klink algorithm that addresses these limitations and provides a much better performance than Klink. Klink-2 introduces a number of new features, including:

- The ability to take as input any kind of statistical or semantic relationship between scholarly keywords and other entities – e.g., authors, organizations, venues and others.
- The ability to handle ambiguous keywords characterized by a noisy set of relationships – e.g., "java", by splitting them into multiple topics and labeling them correctly with their highest level super topic – e.g., "java (programming)" and "java (Indonesia)".
- The ability to scale up to large interdisciplinary ontologies, by being able to generate the topic ontology incrementally on different runs, rather than having to process all the data at the same time.

In the rest of the paper we will describe Klink-2 in detail, illustrating the main features of the algorithm and analyzing its performance in comparison to a number of alternative algorithms. In particular, we will show that the ability of Klink-2 to integrate a high number of data sources and to generate topics with accurate contextual meaning yields significant improvements over the other tested algorithms in terms of both precision and recall.

2 The Klink-2 Algorithm

2.1 Data Model

Many classifications of research areas simply take in consideration a single hierarchical relation, for example the 2012 ACM Classification uses *skos:narrower* to build a taxonomy of topics in computer science. However, as we discussed in [11], this is a limited solution and therefore our model[12], which builds on the BIBO ontology[13], uses a richer set of relationships:

1) *skos:broaderGeneric*. This is used when we have solid evidence that a topic is a sub-area of another one – e.g., "linked data" is a sub-area of "sematic web".
2) *contributesTo* (sub-property of *skos:related).* This indicates that while a topic, *x*, is not a sub-area of another one, *y*, its research outputs contribute to research in *y* to the extent that, for the purposes of querying and exploration, it is useful to consider *x* as 'under' *y*. For example, research on "ontology" contributes to research on "semantic web".
3) *relatedEquivalent* (sub-property of *skos:related*). This indicates that two topics can be treated as equivalent for the purpose of exploring research data – e.g., "ontology mapping" and "ontology matching".

Skos:broaderGeneric and *relatedEquivalent* are necessary to build a taxonomy of topics and to handle different labels for the same research areas, while *contributesTo* provides an additional relationship that can be used to assist the user in browsing research topics [5] and analyzing research data –e.g., for identifying topic-based research communities [10].

2.2 Overview of Klink-2

Klink-2 takes as input a set of scholarly keywords and their relationships with a variety of entities, including research papers, venues, authors, and organizations. The output is a populated OWL ontology describing the semantic relationships between the research topics identified from the set of keywords and the other data provided as input. This semantic network can then be used for improving the processes of searching and performing analytics on scholarly data [3, 5, 6, 7]. As in the case of the Klink algorithm, Klink-2 generates an ontology of research topics linked by the three relationships introduced above. To support those scenarios where we simply wish to gen-

[12] http://kmi.open.ac.uk/technologies/rexplore/ontologies/BiboExtension.owl

[13] http://purl.org/ontology/bibo/

erate the topic network relevant to a specific area – e.g., "Semantic Web", Klink-2 can also start from some given seed topics and expand this initial set by inferring their semantic connections with other topics, which in turn become the new seeds. The user can define a number of levels of recursion after which this process will stop.

The relationships taken as input can be either statistical, such as the number of citations received by the papers tagged with keyword *k* in venue *v*, or semantic, such as the *dbpedia-owl:field* relation used in DBpedia for associating fields to researchers. The former can be derived from article metadata, while the latter can be queried via SPARQL from the Linked Data Cloud or other RDF datasets.

While Klink-2 has been designed to generate ontologies of research topics, it can actually be applied to other domains. For example, we have previously shown that Klink could be used to generate ontologies for recommender systems in the gastronomic domain [12].

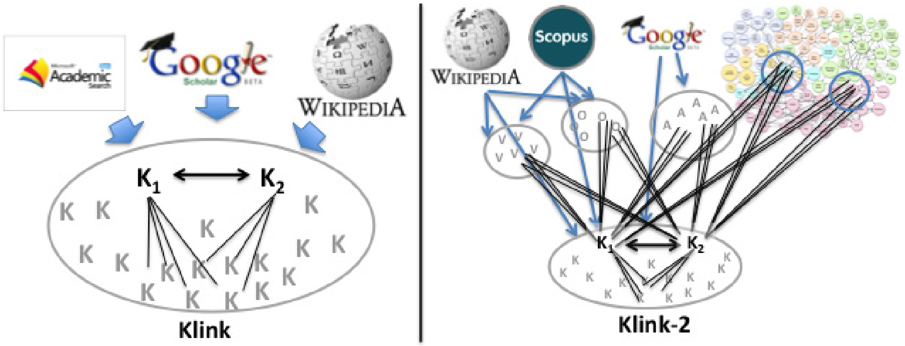


Fig. 1. Relationships used for inferring the topic ontology in Klink and Klink-2.

```
function Klink-2 (keywords, input_rel) returns (owl) {
split_merge=true;
  while (some keywords yet to process) {
   foreach k in keywords {
    keywords2 = getRelatedKeywords(k, input_rel);
     foreach k2 in keywords2 {
       rel = inferRelationships(k,k2, input_rel, rel); }
   }
   rel = fixLoops(rel);
   if (split_merge) keywords = splitAmbiguosKeywords(keywords, rel);
   else keywords = mergeSimilarKeywords(keywords, rel);
   split_merge = ¬ split_merge;
  }
keywords= filterNotAcademicKeywords(keywords, input_rel, rel);
return generateSemanticRelationships(keywords, rel);
}
```

Algorithm 1. The Klink-2 algorithm

Figure 1 shows the difference between Klink and Klink-2 in terms of relationships processed to create the topic network. Klink integrates a number of external sources, but only in order to produce an unbiased co-occurrence graph, which is the only knowledge used by the inference process. Klink-2 can instead exploit multiple relationships and thus take advantage of the rich network of interconnections between the different types of research entities, including papers, authors, venues, and technologies.

The Klink-2 algorithm is structured as follows:

1. Each pair of keywords whose number of common relationships with other scholarly entities is higher than a threshold is analyzed to check whether a hierarchical relationship between the components of the pair can be inferred. If this is the case, *skos:broaderGeneric* and *contributesTo* relationships are derived.
2. Each keyword is analyzed in order to detect possible multiple meanings associated to it. The keywords that seem ambiguous are split into multiple topics with unique meaning, which are then compared to the other keywords, possibly inferring new relationships.
3. The keywords which appear to be very similar are merged together and the *relatedEquivalent* semantic relationships are inferred. As in the previous case, the aggregated keywords are then compared to the already computed ones.
4. Step 2 and 3 are repeated until no new keywords are split or aggregated. Then Klink-2 filters out the keywords that do not represent research areas, fixes the loops in the topic network, and generates the triples describing the semantic relationships between topics.

In what follows we will describe the different phases of the algorithm. We will discuss only briefly the steps already present in the original Klink algorithm – e.g., filtering out keywords which do not denote research areas, to focus instead on the novel solutions.

2.3 Inferring Semantic Relationships

Klink-2 examines each pair of keywords which share a minimum number of relationships to the same scholarly entities and infers the semantic relationships discussed in Section 2.1 by means of three metrics: i) $H_R(x,y)$, which uses a semantic variation of the subsumption method to estimate whether a hierarchical relationship exists between two topics; ii) $T_R(x,y)$, which uses temporal information also to estimate whether a hierarchical relationship exists between two topics; and iii) $S_R(x,y)$, which estimates the similarity between two topics. The first two are used as statistical indicators to detect *skos:broaderGeneric* and *contributesTo* relationships, while the other is used to infer *relatedEquivalent* relationships.

These metrics are computed for each semantic or statistical relation R linking keywords x and y to a set of entities. The keywords (e.g., “semantic web”) are mapped to entities (e.g., *dbpedia:Semantic_Web*) by using DBpedia spotlight[14]. Of course, the

[14] spotlight.dbpedia.org

selected relationships should have a minimum degree of quality and number of linked entities to be analyzed statistically. Hence, in some cases, it can be convenient to aggregate a number of similar semantic relations. For example, DBpedia uses a variety of different relations to connect topics to prominent authors in a discipline, such as *dbpprop:field*, *dbpprop:fields*, *dbpedia-owl:knownFor*. We can thus consider these relations as equivalent for our purposes, so as to improve the number of linked entities and the robustness of the statistical inferences.

2.3.1 Hierarchical Relationship Indicators

A classical way to infer a hierarchical relationship between two entities, which can occur in a set of documents, is the subsumption method [13]. According to this approach, term x subsumes term y if $P(x|y) \geq \alpha$ and $P(y|x) < 1$, with α usually set to 0.8. The original Klink improved on this method by considering the similarity between the distributions of co-occurring keywords as well as their string similarity. Klink-2 generalizes this approach by taking also in consideration the relationships linking keywords x and y to common entities. It does it by computing the conditional probability that an entity e linked to x by relation R will also be linked to y by the same relation. For example, a relationship between "semantic web" and "linked data" can also be inferred by the probability that an author working in one of these topics would also work in the other, or that a tool used in one of these topics would be used in the other. Hence, for every relation R, Klink-2 computes two statistical indicators ($H_R(x,y)$ and $T_R(x,y)$) that are used to detect a hierarchical relationship and then establish its nature.

Our approach distinguishes two classes of relations: quantified and unquantified ones. An unquantified relation is a triple in the form of *rel(t, e)* linking a topic t to an entity e. For instance, this could be a triple of the form *isAbout(p, t)* from the SWRC ontology, which states that a publication p is about topic t. A quantified relation is a quadruple in the form of *rel(t, e, q)*, where q quantifies numerically the intensity of the relationship. For example, *haveCitationInTopic*(a, t, 25) points to the fact that author a has 25 citations in topic t. The former are usually queried directly from RDF repositories, while the latter are inferred from metadata.

Using these input data we compute the statistical indicator $H_R(x,y)$ between keywords x and y for relation R with the following formula:

$$H_R(x,y) = \left(\frac{I_R(x,y)}{I_R(x,x)} - \frac{I_R(y,x)}{I_R(y,y)}\right) \cdot c_R(x,y) \cdot n(x,y) \tag{1}$$

The first factor gives the direction of the possible hierarchical relationship, while the others give the intensity. $\frac{I_R(x,y)}{I_R(x,x)}$ is the conditional probability that an element associated with keyword x will be associated also with keyword y. If R is an unquantified relation, $I_R(x,y)$ is simply the number of elements associated with both x and y according to relation R. For example, in the case of *isAbout(p, x)*, $I_R(x,y)$ is equal to the number of co-occurrences between x and y, while $I_R(x,x)$ and $I_R(y,y)$ indicate the total number of publications in x and y. If R is a quantified relation, we should also take into account the intensity of the relationship. In this case, $I_R(x,y)$ is computed as the summation of the minimum values quantifying the two relationships connecting

x and y with e. For example, in the case of the relationship *haveCitationInTopic(a,x,c)*, $I_R(x,y)$ is the sum of the minimum numbers of citations in x and y received by each author, while $I_R(x,x)$ and $I_R(y,y)$ are respectively the sum of the total number of citations in x and in y received by all authors.

$c_R(x,y)$ measures the semantic similarity of x and y and is computed as the cosine similarity between the two vectors in which each index represents the keyword k, which has in common with x and/or y a set of instantiations of a relation, say R, with the same scholarly entities, with the values equal to $I_R(k,x)$ for x and $I_R(k,y)$ for y.

Finally $n_R(x,y)$ defines the string similarity between two keywords. It is computed as the linear combination of a number of string metrics based on the longest common sub-string, the percentage of identical words, the number of characters in common, the presence of acronyms, and so on.

When $H_R(x,y) \geq t_R$ we infer that, according to relation R, x is a candidate to becoming a sub-area of y, while when $H_R(x,y) \leq -t_R$, x is a candidate to becoming a super-area of y. The value of t_R can be set manually by analyzing the trade-off between precision and recall or alternatively it can be estimated by running the algorithm on training data and using the Nelder-Mead algorithm [12] to choose the thresholds which maximize the performances (usually in term of F-measure).

It is interesting to note that the formula used by the original Klink algorithm [11] can be considered (except for the improved $n_R(x,y)$ component) as a $H_R(x,y)$ indicator, using as relation *isAbout(p,x)*.

In many cases, it is also useful to consider the diachronic component of the relationships between two keywords, e.g. how their relationship evolved in time. For example, in the case of *isAbout(p,x),* it can be argued that after some time certain topics may stop to co-occur simply because their association has become implicit.

This may cause a statistical indicator, which does not consider the diachronically dimension, to miss some important semantic relationships. Moreover the temporal dimension is useful to understand better the nature of the relationship linking two topics. The fact that the relationship was strong when one of the topics was young may point to the fact that this topic actually derived from the other and thus is truly one of its sub-areas. For this reason, Klink-2 computes also $T_R(x,y)$, a temporal version of $H_R(x,y)$, which gives more weight to the information associated with the first years of x. This is calculated using a variation of formula (1) in which $I_R(x,y)$ is computed by weighting the number and intensity of the relationships in each year according to the distance from the debut of x. The weight is computed as *w(year, x)= (year - debut(x) +1)*$^{-\gamma}$, with $\gamma>0$ ($\gamma=2$ in the prototype).

2.3.2 Inferring Hierarchical Semantic Relationships

A hierarchical relationship between two topics (represented by the keywords) is inferred when a sufficient number of indicators, i.e., a number above a given threshold, agree on the direction of the relationship. The precise threshold depends on the desired precision/recall trade-off. In some rare cases the situation may arises where indicators provide conflicting information – i.e., both $x > y$ and $y > x$ are suggested. In such a case we compute the difference between the two groups and go for a ‘majority vote’, assuming the difference is higher than the given threshold.

The nature of the inferred relationship is assessed by Klink-2 using a rule-based approach. This method takes into consideration a variety of factors, including the number of publications associated to x and y, the number of entities related to them, their debut years (i.e., the years in which the keywords first appeared), and the prevalence of $T_R(x, y)$ indicators versus $H_R(x, y)$ ones. If x is older, associated with more entities and there is a prevalence of $T_R(x, y)$ indicators, Klink-2 will infer a *skos:broaderGeneric* relationship. If these conditions do not apply, it will infer a *contributesTo* relationship. If the choice is unclear, it will be conservative and generate a *contributesTo* relationship since it provides a less risky assumption. A *skos:broaderGeneric(x,y)* relationship is transitive and implies that every publication tagged with x should also be tagged with y. Hence it is important to minimize as much as possible errors with the derivation of *skos:broaderGeneric* relationships, which will adversely affect the exploration of the scholarly data.

At the end of each main analysis loop, Klink-2 will also run the `fixLoops()` procedure, which detects loops in the graph of *skos:broaderGeneric* relationships and breaks them by eliminating the relationships with weaker statistical indicators.

2.3.3 Inferring RelatedEquivalent Relationships

Klink-2 uses the $S_R(x, y)$ similarity metric to infer *relatedEquivalent* relationships. We compute $S_R(x, y)$ by normalizing $c_R(x, y)$ with respect to the similarity between the super-areas and the siblings of x and y, according to the previously inferred hierarchical relationships. For this reason the *relatedEquivalent* relationships start to be inferred only after the first loop. The rationale is that for considering two elements in a taxonomy near enough to be merged they must be not only similar in absolute terms, but also more similar to each other than their super areas and siblings are to each other. Hence, we adopt the following formula:

$$S_R(x, y) = \frac{c_R(x,y)}{max\left(c_R{}^{super}(x,y)\,, c_R{}^{sib}(x,y)\right)+1} \tag{2}$$

This formula is an evolution of the one used in Klink and proved to work better both on scholarly domains and on other domains [12]. Each pair of keywords which receives enough positive indicators is then linked by a similarity link. These pairs are then given in input to a bottom-up single-linkage hierarchical clustering algorithm [14], labeled in the pseudocode as `mergeSimilarKeywords()`, which uses as distance criterion a linear combination of the $S_R(x, y)$ indicators. For each pair of keywords clustered together, Klink-2 infers a *relatedEquivalent* relationship. The keywords in the cluster are then merged by aggregating all their relationships and will be re-analyzed in the next loop to infer additional relationships

2.4 Handling Ambiguous Keywords

The assumption that each keyword can be mapped to only one topic is unsafe, even when we consider keywords which were directly associated to a paper by the authors themselves. Our analysis on a subset of the Scopus dataset revealed mainly three categories of ambiguous keywords:

1. Terms which happen to have two or more different meanings, e.g., "java", the programming language, and "java", the island.
2. Vague terms, with meaning that can change according to the paper they are associated to – e.g., "mapping".
3. Terms that used to have a unique meaning, but are now used in specialized ways by different research communities – e.g. "ontology".

The first case is the most trivial, but also the one that may yield the biggest mistakes. For example, the original version of Klink, when processing a mixed database of life science and computer science, would infer that "owl" is both a sub-area of "semantics" and of "birds". The second case is partially addressed by the original Klink by excluding from the process the generic terms that co-occur significantly with a very high number of uncorrelated keywords. However, this quick solution may lose potentially interesting pieces of information. For example, we may assume with a good degree of confidence that the keyword "mapping", when combined with "ontology" and "interoperability", acquires an accurate meaning that is useful to capture. The third category is subtler, but can still yield a number of problems both for users, who may want to query the data using only the meanings more commonly used in their research community, and for algorithms that rely on statistical inferences. For example, "ontology" is used by most philosophers with the original meaning of study of the nature of being, while computer scientists usually refer to it as a practical tool for modeling a domain.

The ambiguous keywords are usually associated with a noisy set of relationships, which hinders the statistical inference process discussed in section 2.3. For this reason, Klink-2 addresses these cases by detecting the ambiguous terms and splitting them in multiple distinct topics. Differently from the disambiguation of probabilistic topic models [15, 16, 17], this process is driven by both pre-existing and inferred semantic relationships.

```
function splitAmbiguosKeywords(keywords, rel) returns (keywords) {
   foreach k in keywords {
      related_keywords = getRelatedKeywords(keywords, rel);
      clusters = quickHierarchicalClustering(related_keywords, rel);
      if ( count(clusters) > 1) {
         clusters2 = intersectBasedClustering(related_keywords, rel);
         if ( count(clusters2) > 1) {
            keywords = split(k, clusters2, keywords, rel); }
      }
   }
return keywords;
}
```

Algorithm 2. Detecting and splitting ambiguous keywords.

The first step is to quickly detect that a keyword *x* is probably ambiguous and thus a valid candidate to be analyzed more in depth. Since Klink-2 aims to be a scalable method, able to process a very large number of keywords, this first phase should be as quick as possible. To this purpose, we first select the keywords which share with x a minimum number of relationships to the same entities. We then run a hierarchical bottom-up clustering algorithm on this set of keywords, using as initial distance a linear combination of the $S_R(x, y)$ indicators. At each iteration of the algorithm, the distances between the new cluster n and each other cluster c is quickly updated by computing the weighted average of the distances between the merged elements and c, using as weight the number of papers associated with each keyword. If the algorithm yields more than one cluster, Klink-2 estimates that the analyzed keyword is connected to two or more distinct groups of keywords and thus may be ambiguous. For example, the keywords associated to 'owl' would be grouped in two clusters, one including terms such as 'RDF and 'semantic web' and the other including terms such as 'raptores' and 'barn owl'. However, it would be careless to directly generate new topics from this result, since a keyword may actually be associated with different groups of keywords without necessary being ambiguous. For this reason we run a slower and more accurate clusterization algorithm only on the keywords that yielded more than one cluster in the first phase. This method, `intersectBasedClustering()`, assigns to each cluster a pseudo-keyword, whose relationships are recomputed by considering only the entities that are connected both with the potential ambiguous keyword and at least one of the other keywords occurring in the cluster, which thus act as disambiguators. For example, in the case of "owl", the *isAbout* relation will be recomputed by considering only the publications tagged by the intersection of "owl" and a number of keywords associated to the general meaning of either "semantics" or "birds". The clustering process is then restarted and, at each iteration, the distances between clusters are re-calculated by updating the pseudo-keywords. If the process yields more than one cluster, the original keyword is used to produce as many topics as the resulting number of clusters. This is done by inserting the pseudo-keywords associated with the final clusters in the set of keywords to analyze, after labeling them accordingly to the most important high-level topics in the cluster. The related higher-level keyword used in the label is the member of the cluster with the highest harmonic mean between the number of co-occurrences with the original keyword and its total number of associated publications. For example, "owl" may be split into two different pseudo-keywords: "*owl (semantics)*" and "*owl (birds)*". These keywords will be associated with the set of disambiguated relationships re-computed during the clustering process and will be compared with the other keywords for inferring new relationships.

In some cases, it would be inconvenient for the algorithm to return all the possible meanings of a keyword. For example, a researcher interested in the Semantic Web would just want the algorithm to automatically assign to "owl" the meaning of "owl (semantics)", without actually producing a second topic related to birds. For this reason, the approach can also be run in *contextual mode*. In this modality, Klink-2 will only keep the disambiguated keyword that is more similar to the input keywords, according to the cosine distance of the associated keyword distributions. Hence, if the input keywords were about the Semantic Web, "owl" will automatically take the correct contextual meaning and have its relationships disambiguated by using keywords about "semantics".

The threshold to stop the clustering process can be set to a high value, so to address only the first two categories of ambiguous keywords, or can be relaxed to tackle also the third one. While the second solution may produce an excessively fine-grained set of topics, it will also reduce the noise in the data and foster the quality of the relationships, by mapping each topic to a very accurate and unique meaning.

2.5 Triple Generation

Klink-2 exits the main loop when it has no more keywords to analyze. It then filters the keywords considered "not academic" or "too generic" according to a number of heuristics, such as the profile of distribution of their co-occurrences or their absence from relevant academic sources – this process is fully described in [11]. While the first version of Klink used to filter the keywords before analyzing them, Klink-2 does it afterwards. This is because the ability to process ambiguous keywords can actually generate usable topics from many of the keywords that the original version would have discarded. In this phase, Klink-2 also deletes the redundant relationships which would be entailed by other relationships. Finally, Klink-2 generates the triples describing the research topics and their relationships. The output can be used to create a new OWL knowledge base or can be added to an existing one. In the latter case Klink-2 will check the relationships for inconsistencies and loops and may delete some of them. Being able to build an ontology iteratively on different runs is indeed very useful to address scalability, since the algorithm will not be forced to load the full graph of all existing keywords, but can run on different sub-taxonomies, which are then merged.

3 Evaluation

We tested our approach on the keywords of a dataset extracted from Scopus, consisting of 16 million publications about computer science and life sciences. Additional knowledge about these keywords and their relationships was extracted from DBpedia, Google Scholar and Wikipedia. We evaluated our method by testing a number of alternative algorithms for their ability of building an ontology about the Semantic Web and related areas. To this end, we adopted as gold standard the ontology used in [11], after updating it by i) mapping some of the terms in the ontology to keywords used by Scopus (e.g., "linked datum"), which were not present in the data used in the 2012 evaluation, and ii) adding 30 new topics co-occurring with "Semantic Web" and "Semantics" in the Scopus database. The new version of the ontology was validated and corrected by three external domain experts with publications in ISWC and ESWC conferences. The resulting gold standard[15] includes 88 topics linked by 133 semantic relationships (263 when taking in consideration also the subsumption relationships that can be derived from transitive relations).

[15] The gold standard and the data generated in the evaluation are publicly available at http://kmi.open.ac.uk/technologies/rexplore/iswc2015/.

We tested four different methods:

1) the classic subsumption method [8, 13], mentioned in section 2.3.1 (labelled **S**);

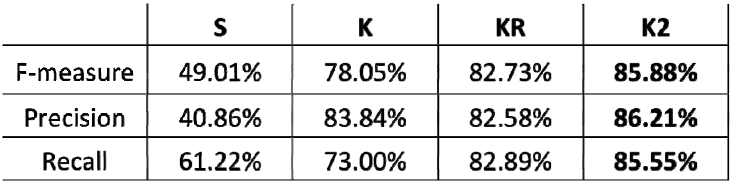
2) the original Klink algorithm, as described in [11] (labelled **K**);

3) a first version of Klink-2, with the ability of integrating multiple relationships, but not addressing ambiguous keywords (labelled **KR**);

4) the final version of Klink-2, with also the ability to detect and split ambiguous keywords in contextual mode (labelled **K2**);

The co-occurrence graph derived from Scopus was enriched by exploiting the co-occurrences on Google Scholar and Wikipedia, as described in [11]. **KR** and **K2** used six statistical relationships computed on the Scopus dataset, i.e. the number of associated publications/citations for publications, authors and venues. These methods also queried a variety of semantic relationships from DBpedia, such as *foaf:primaryTopic*, *dbpprop:discipline*, *dcterms:subject*, *dbpprop:domain*, *dbpprop:field*, *dbpedia-owl:knownFor* and so on. The thresholds for **S, K**, **KR** and **K2** were set to maximize the F-measure on the topic taxonomy used by Rexplore [5], and originally generated from the Microsoft Academic Search dataset. The minimum number of indicators used by **KR** and **K2** for inferring semantic relationships was empirically set to 2.

Table 1. F-measure, precision and recall of the four approaches.



	S	K	KR	K2
F-measure	49.01%	78.05%	82.73%	**85.88%**
Precision	40.86%	83.84%	82.58%	**86.21%**
Recall	61.22%	73.00%	82.89%	**85.55%**

The ontologies generated by **S**, **K**, **KR** and **K2** were compared with the gold standard by computing recall, precision and F-measure of the inferred semantic relationships. Table 1 shows the metrics relative to the four approaches. The statistical significance between the approaches was assessed by arranging data in cross-correlation tables analyzed with the chi-square test (with Yates' correction for 2x2 tables). All outcomes of **K, KR** and **K2** are significantly superior to those of **S** ($p < 0.0001$), confirming the results presented in [11]. The F-measure increases from **K** (78%) to **KR** (83%), to **K2** (86%), with a significant difference between **K2** and **K** (p=0.001). The precision is essentially similar for **K** and **KR,** improving slightly for **K2** (p=0.51). However, the recall increases notably from **K** (73%) to **KR** (83%) to **K2** (86%) with differences which are significant for **KR** versus **K** (p=0.008) and even more so for **K2** versus **K** (p=0.0005).

Hence, the results indicate that allowing the approach to take into account multiple relationships has an important impact on the recall of semantic relationships. Moreover, the technique to address ambiguous keywords discussed in section 2.4 yields a significant improvement in both precision and recall.

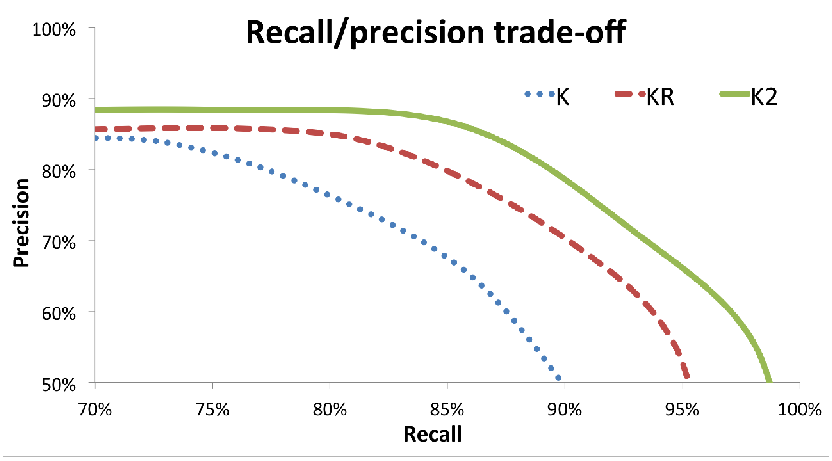


Fig. 2. Recall/precision trade off.

In many scenarios, the users may want to optimize the approach so that it yields either a high recall or a high precision, depending on the context. For example, if humans will validate and correct the generated semantic relationships, e.g., by using crowdsourcing ontology verification [18, 19], it may be more important to have a high recall. On the contrary, if this step is not carried out and the ontology is used by automatic methods, precision is usually more important.

We explored the recall/precision trade-off of **K**, **KR** and **K2** by running the algorithms with different thresholds modulated by a factor ranging from 0.25 to 3, to obtain an increasingly stricter inference process. Figure 3 shows the precision of the algorithms as a function of the recall. **K2** clearly outperforms again the other two algorithms by yielding a higher precision over the whole recall range (p=0.005 with non-parametric Wilcoxon's test), especially in the highest recall region. For example, when a recall of 90% is required, **K** yields a precision of 50%, **KR** of 73% and **K2** of 80%: hence, Klink-2 allows also a greater flexibility in choosing the recall/precision trade-off tailored to the user needs. Taking in consideration a high number of relationships obviously requires more time. The topics of the gold standard were analyzed in about 4 seconds (average on the various runs) by **S**, in 7 by **K**, in 36 by **KR** and in 45 by **K2**. However, since this kind of algorithm does not usually run in real time (e.g., Rexplore updates its ontology every three months), an increment in running time is a low price to pay for significantly better performances.

4 Related Work

Ontologies of research topics can be helpful for exploring and making sense of academic data in a variety of ways. For example, they can enhance semantically many information extraction techniques, such as trend detection [6] and community detection [7, 10]. They also make it possible to improve search results and their presentation, e.g., by supporting semantic faceted search [20].

There are a variety of approaches for learning taxonomies or ontologies, including natural language processing [21], clustering techniques [22], statistical methods [13], and methods based on spreading activation [19]. Text2Onto [21] is a popular system for learning ontologies, which represents the learned ontological structures in a probabilistic ontology model and uses natural language processing techniques. The Lexico-Syntactic Pattern Extraction (LSPE) approach [23] exploits linguistic patterns, e.g., "such as…" and "and other…", to discover relationships between terms. However, these approaches are based on the analysis of textual documents, while Klink-2 focuses instead on metadata, statistics and semantic relationships, since its scope is a large-scale analysis of research data.

The *TaxGen* framework [14] creates taxonomies from a set of documents by means of a hierarchical agglomerative clustering algorithm and text mining techniques. Klink-2 also adopts a clusterization algorithm for inferring the *relatedEquivalent* relationship and handling ambiguous keywords.

A very popular statistical approach is the subsumption method [13], which computes the conditional probability for a keyword to be associated with another in order to infer hierarchical relationships, as discussed in section 2.2. The same idea is extended in the GrowBag algorithm [8], which enriches the original model by using second order co-occurrences made explicit by a biased *PageRank* algorithm. The original Klink algorithm [11] also used statistical methods on the co-occurrence graph, while Klink-2 goes a step further by allowing the use of semantic or statistical relationships from multiple sources. The use of multiple sources for this task was also strongly advocated by Wohlgenannt et al [19], who proposed a framework for inferring lightweight ontologies which first build a semantic network through co-occurrence analysis, trigger phrase analysis, and disambiguation techniques, and then uses spread activation to find candidate concepts. Klink-2 does a similar co-occurrence analysis, but also uses indirect relationships and generates novel topics derived from the combination of different keywords. Similarly to the approach of Wohlgenannt et al, Klink-UM [12], a variation of Klink designed to generate lightweight ontologies for recommender systems, adopts spreading activation for tailoring semantic relationships to user needs.

Klink-2 is able to manage ambiguous keywords by generating multiple topics with a unique meaning, according to the semantic context. This is conceptually similar to the disambiguation performed by probabilistic topics models which detect latent topics by exploiting Probabilistic Latent Semantic Indexing (pLSI) [15] or Latent Dirichlet Allocation [16]. For example the Author-Conference-Topic (ACT) model [17] treats authors as probability distributions over topics, conferences and journals. Differently from them, our approach uses explicit semantic relationships, rather than latent semantic, to drive the generation of unambiguous topics. These topics are accurately described by a number of semantic relationships and not simply as term distributions.

Methods for automatically learning ontologies can be complementary to crowdsourcing ontology verification [18, 19], a process in which a large number of workers solve micro-tasks for validating and correcting semantic relationships.

As mentioned in the introduction, Klink-2 is currently integrated in the Rexplore system [5], and is used to semantically enhance a number of algorithms for exploring research data. Nowadays we have several interesting tools which exploit semantic technologies to make sense of research. The Saffron system [9], which builds on the Semantic Web Dog Food Corpus [1], allows for advanced expert search and estimates the strength of an author/topic relationship by analyzing co-occurrences on the Web. Arnetminer [17] also provides support for expert search and a variety of analytics on research topics. RKBExplorer [3] is an application that generates comprehensive visualizations of the research environment from a number of heterogeneous data sources. Klink-2 can benefit these systems by generating an accurate, large-scale and up-to-date topic network.

5 Conclusions

We presented Klink-2, a novel approach to generate semantic topic networks which can integrate a number of web sources and exploit multiple semantic and statistical relationships. The output can be useful to a vast number of tools as it can be used to provide a semantic structure to support the identification, search, exploration and visualization of research data. The evaluation shows that Klink-2 performs significantly better than alternative solutions. In particular, Klink-2 is able to yield a good precision (80%) even when a very high recall (90%) is needed.

Our approach opens up many interesting directions of work. On the research side, we plan to investigate diachronically the shift in meaning of scholarly keywords to better characterize the evolution of research areas. We also want to exploit natural language processing techniques to augment our semantic model with additional entities (e.g., methods, tools, and standards) which can be extracted from the text of scientific publications. Finally, on the technology transfer side, we are currently collaborating with two major academic publishers, who are looking to deploy Klink-2 in their organizations, thus providing a strong semantic topic structure to support classification, search and exploration in their digital libraries.

Acknowledgements. We would like to thank Elsevier BV and Springer DE for providing us with access to their large repositories of scholarly data.

References

1. Moller, K., Heath, T., Handschuh, S., Domingue, J.: Recipes for semantic web dog food — the ESWC and ISWC metadata projects. In: 6th International Semantic Web Conference, November 11–15, 2007, Busan, South Korea (2007)
2. Latif, A., Afzal, M.T., Helic, D., Tochtermann, K., Maurer, H.: Discovery and construction of authors' profile from linked data (A case study for Open Digital Journal). In: WWW 2010 Workshop on Linked Data on the Web (LDOW 2010). CEUR-WS, vol. 628, Raleigh, North Carolina, USA (2010)
3. Glaser, H., Millard, I.: Knowledge-enabled research support: RKBExplorer.com. In: Proceedings of Web Science 2009, Athens, Greece (2009)
4. Peroni, S., Shotton, D.: FaBiO and CiTO: ontologies for describing bibliographic resources and citations. Journal of Web Semantics **17**, 33–43 (2012)

5. Osborne, F., Motta, E., Mulholland, P.: Exploring scholarly data with rexplore. In: Alani, H., et al. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 460–477. Springer, Heidelberg (2013)
6. Decker, S.L., Aleman-Meza, B., Cameron, D., Arpinar, I.B.: Detection of bursty and emerging trends towards identification of researchers at the early stage of trends (Doctoral dissertation, University of Georgia) (2007)
7. Erétéo, G., Gandon, F., Buffa, M.: SemTagP: semantic community detection in folksonomies. In: 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 324–331. IEEE (2011)
8. Diederich, J., Balke, W., Thaden, U.: Demonstrating the semantic GrowBag: automatically creating topic facets for FacetedDBLP. In: JCDL 2007, NY, USA (2007)
9. Monaghan, F., Bordea, G., Samp, K., Buitelaar, P.: Exploring your research: sprinkling some saffron on semantic web dog food. In: Semantic Web Challenge at the International Semantic Web Conference (2010)
10. Osborne, F., Scavo, G., Motta, E.: Identifying diachronic topic-based research communities by clustering shared research trajectories. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 114–129. Springer, Heidelberg (2014)
11. Osborne, F., Motta, E.: Mining semantic relations between research areas. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 410–426. Springer, Heidelberg (2012)
12. Osborne, F., Motta, E.: Inferring semantic relations by user feedback. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS, vol. 8876, pp. 339–355. Springer, Heidelberg (2014)
13. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proceedings of the SIGIR Conference, pp. 206–213 (1999)
14. Müller, A., Dorre, J.: The TaxGen framework: automating the generation of a taxonomy for a large document collection. In: Proceedings of the 32nd Hawaii International Conference on System Sciences, vol. 2, pp. 20–34 (1999)
15. Hofmann, T.: Probabilistic latent semantic indexing. In: the 22nd Conference on Research and Development in Information Retrieval, pp. 50–57, Berkeley, CA (1999)
16. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research **3**, 993–1033 (2003)
17. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining, pp. 990–998. ACM (2008)
18. Mortensen, J.M., Alexander, P.R., Musen, M.A., Noy, N.F.: Crowdsourcing ontology verification. In: The Semantic Web–ISWC 2013, pp. 448–455 (2013)
19. Wohlgenannt, G., Weichselbraun, A., Scharl, A., Sabou, M.: Dynamic integration of multiple evidence sources for ontology learning. Journal of Information and Data Management **3**(3), 243 (2012)
20. Suominen, O, Viljanen, K., Hyvänen, E.: User-centric faceted search for semantic portals. In: 4th European Conference on the Semantic Web (ESWC 2007), pp. 356–370 (2007)
21. Cimiano, P., Völker, J.: Text2Onto. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg (2005)
22. Assadi, H.: Construction of a regional ontology from text and its use within a documentary system. In: Guarino, N. (ed.) Proceedings of FOIS 1998 Formal Ontology in Information Systems, pp. 236–249, Trento, Italy (1999)
23. Hearst, M.: Automated discovery of WordNet relations. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 131–153. MIT Press (1998)