Chapter 2
Automatic Invocation Linking for Collaborative Web-Based Corpora

James Gardner, Aaron Krowne, and Li Xiong

Abstract Collaborative online encyclopedias or knowledge bases such as Wikipedia and PlanetMath are becoming increasingly popular because of their open access, comprehensive and interlinked content, rapid and continual updates, and community interactivity. To understand a particular concept in these knowledge bases, a reader needs to learn about related and underlying concepts. In this chapter, we introduce the problem of invocation linking for collaborative encyclopedia or knowledge bases, review the state of the art for invocation linking including the popular linking system of Wikipedia, discuss the problems and challenges of automatic linking, and present the NNexus approach, an abstraction and generalization of the automatic linking system used by PlanetMath.org. The chapter emphasizes both research problems and practical design issues through discussion of real world scenarios and hence is suitable for both researchers in web intelligence and practitioners looking to adopt the techniques. Below is a brief outline of the chapter.

Problem and Motivation. We first introduce the problem of invocation linking for online collaborative encyclopedia or knowledge bases. An online encyclopedia consists of multiple entries. An *invocation link* is a hyperlink from a term or phrase in an entry representing a concept to another entry that defines the concept. It allows a reader easily "jump" to requisite concepts in order to fully understand the current one. We refer to the term or phrase being linked from as *link source* and the entry being linked to as *link target*. The problem of *invocation linking* is how to add these invocation links in an online encyclopedia in order to build a semantic concept network.

State of the Arts. We review the state of arts for the invocation linking in current online encyclopedia and knowledge bases. The existing approaches can be mainly

J. Gardner (✉) and L. Xiong
Department of Mathematics and Computer Science, Emory University, 400 Dowman Dr. Atlanta, GA 30322
e-mail: jgardn3@emory.edu; lxiong@mathcs.emory.edu

A. Krowne
PlanetMath.org, 4336 Birchlake Ct. Alexandria, VA 23309
e-mail: akrowne@gmail.com

R. Chbeir et al., *Emergent Web Intelligence: Advanced Information Retrieval*, Advanced Information and Knowledge Processing, DOI 10.1007/978-1-84996-074-8_2,

classified into: 1) *manual linking* where both the link source and link target are explicitly defined by the user (such as blog software), 2) *semi-automatic linking* where the link source are explicitly marked by the user but the link target is determined automatically (such as Wikipedia), and 3) *automatic linking* where both the link source and link target are determined automatically. We discuss the representative systems for each approach and illustrate their advantages and disadvantages. We will also review potential technologies such as web search and recommender systems and discuss their applicability for invocation linking.

Automatic Invocation Linking. We advocate in this chapter the automatic linking approach as we believe that the manual and semi-automatic approaches are an unnecessary burden on contributors, and in addition, require continuous re-inspection of the entire corpus by writers or other maintainers for a growing and dynamic corpus. We discuss the challenges and design goals for developing such an automatic linking system including linking quality, efficiency and scalability, and generalization to multiple corpus.

NNexus Approach. In particular, we present the NNexus system, an automatic linking system that we have developed as an abstraction and generalization of the linking component of PlanetMath (planetmath.org), PlanetPhysics(planetphysics.org), and other sites. We discuss a number of key features and design ideas of NNexus in addressing the challenges for invocation linking. NNexus provides an effective linking scheme utilizing metadata to automatically identify link sources and link targets. It achieves good linking quality with a classification-based link steering approach and an interactive entry filtering component. It achieves good efficiency and scalability by its efficient data structures as well as a mechanism for efficiently updating the links between entries that are related to newly defined or modified concepts in the corpus. Finally, its implementation utilizes OWL and has a simple interface, which allows for an almost unlimited number of online corpora to interconnect for automatic linking.

Conclusions and Open Issues. We close the chapter by discussing a set of interesting issues and open problems for invocation linking.

2.1 Introduction

Collaborative online encyclopedias or knowledge bases such as Wikipedia[1] and PlanetMath[2] are becoming increasingly popular because of their open access, comprehensive and interlinked content, rapid and continual updates, and community interactivity.

To understand a particular concept in these knowledge bases, a reader needs to learn about related and underlying concepts. Thus, a knowledge based should

[1] http://www.wikipedia.org

[2] http://www.planetmath.org

contain the appropriate links for all of the concepts to the appropriate definitions or articles. These links should allow browsing to all the concepts that are evident to the reader's intuition.

The popularity of these encyclopedic knowledge bases has also brought about a situation where the availability of high-quality, canonical definitions and declarations of educationally useful concepts have outpaced their usage (or *invocation*) in other educational information resources on the web. Instead, the user must execute a new search (either online or offline) to look up an unknown term when it is encountered, if it is not linked to a definition. For example, blogs, research repositories, and digital libraries quite often do not link to definitions of the concepts contained in their texts and metadata, even when such definitions are available. This is generally not done because of the lack of appropriate software infrastructure and the extra work creating manual links entails. When such linking is actually done, it tends to be incomplete and is quite laborious.

2.1.1 Problem Definition

In this chapter, we study the problem of invocation linking to build a semantic network for collaborative online encyclopedia. We first define a number of terminologies and define our problem to facilitate our discussion.

A *collaborative online encyclopedia* is a kind of knowledge base containing "encyclopedic" (standardized) knowledge contributed by a large number of participants (typically but not necessarily in a volunteer capacity). Any article submitted by a user in such a collaborative corpus is an *entry* or an *object*. We say *invocation* referring to a specific kind of semantic link: that of *concept invocation*. Any statement in a language is composed of concepts represented by tuples of words. Such a statement invokes these concepts, as evidenced by the inclusion of word tuples that correspond to common labels for the concepts. We call these tuples of words *concept labels*. A *invocation link* is a hyperlink from these tuples of words in an entry that represent a concept to an entry that defines the concept. We refer to the tuples of words being linked from as *link source* and the entry being linked to as *link target*. The problem of *invocation linking* is how to add these invocation links in an collaborative online encyclopedia.

The table in Fig. 2.1 shows a list of entries (objects) in an example online encyclopedia[3] corpus with their object ID and metadata including what concepts each entry defines and the Mathematical Subject Classification (MSC) for each entry. It also shows an example entry[4] with links to concepts that are defined in the same corpus. The terms underlined indicate terms that need to be linked based on the meta-data in the table. For example, *planar graph* in the example entry needs to

[3] http://planetmath.org

[4] Extracted from http://planetmath.org/encyclopedia/PlaneGraph.html

ObjectId	Concepts defined	MSC
1	Triangle, right triangle, ...	51-00
2	Planar, planar graph, ...	05C10
3	Connected, ...	05C40
4	Geometry, Euclidean geometry, ...	01A16
5	Graph, graph theory, edge, ...	05C99
6	Graph, function graph	03E20

A planar graph is a graph which can be drawn on a plane (a flat 2-d surface) or on a sphere, with no edges crossing. When drawn on a sphere, the edges divide its area in a number of regions called faces (or "countries", in the context of map coloring). Even if...

Fig. 2.1 Example document corpora with meta-data and example entry

be linked to object (entry) 2 that defines the concept *planar graph*. We will use this example to explain the concepts discussed in this chapter.

While it is possible to extend the problem definition and the techniques we will discuss for other types of linking such as links to articles with a similar or different point of view, it is our focus in this chapter to study *concept* or *definitional* linking.

2.1.2 Chapter Overview

In this chapter, we study the problem of invocation linking for collaborative encyclopedia or knowledge bases, review the state of the art for invocation linking, discuss the problems and challenges of automatic linking, and present the NNexus approach, an abstraction and generalization of the automatic linking system used by PlanetMath.org. The chapter emphasizes both research problems and practical design issues through discussion of real world scenarios and hence is suitable for both researchers in web intelligence and practitioners looking to adopt the techniques. Below is a brief outline of the chapter.

Section 2.2 reviews the state of arts for the invocation linking in current online encyclopedia and knowledge bases. We discuss the representative systems and illustrate their advantages and disadvantages and motivate the automatic linking approach. We will also review potential technologies such as web search, recommender systems and machine learning and discuss their applicability for invocation linking. In Section 2.3, we discuss a set of general challenges and design goals for an automatic linking system to achieve including linking quality, efficiency and scalability, and generalization to multiple corpus. In Section 2.4, the main part of the chapter, we present the NNexus system, an automatic linking system that we have developed as an abstraction and generalization of the linking component of PlanetMath (planetmath.org), PlanetPhysics(planetphysics.org), and other sites [4]. We discuss a number of key features and design ideas of NNexus in addressing the challenges for invocation linking. Finally, we close the chapter in Section 2.6 by discussing a set of interesting issues and open problems for invocation linking.

2.2 State of the Art

We briefly survey the existing and potential solutions for invocation linking and motivate the automatic linking approach. We also review a number of technologies that are related or applicable to the invocation linking problem.

2.2.1 Invocation Linking

The existing and potential approaches for invocation linking can be mainly classified into the following three categories, namely, *manual linking*, *semi-automatic linking*, and *automatic linking*.

2.2.1.1 Manual Linking

Manual linking refers to the linking technique where both the link source and link target are explicitly defined, e.g., anchor tags in html documents. Most web pages use the manual approach. Blog software (such as Wordpress) generally requires writers create links manually.

2.2.1.2 Semi-automatic Linking

Semi-automatic linking refers to the technique where the terms at the source are explicitly marked for linking, but the link target is determined by the collaborative online encyclopedia system. Many current online encyclopedias (including Wikipedia) use the semi-automatic approach.

Wikipedia (which is powered by the Mediawiki software) uses a semi-automatic approach. That is, the links are manually delimited by authors when the author invokes a concept that they believe should be defined in the collection, but the system disambiguates between the possible destinations for the link. If an entry for a concept is present only by an alternate name, the link might fail to be connected. Links to non-existent entries are rendered specially as “broken” links, and the Mediawiki system makes it easy to start a new entry for that term. However, this is inherently somewhat distracting to those uninterested in creating a new entry. Mediawiki and other systems that take a similar approach also fail to provide systemic treatment of homonymy. The Wikipedia convention is to manually create “disambiguation nodes,” which contain links to all homonymous concepts with a particular label. Such nodes add an extra step to navigation, require ongoing maintenance, and can contain an extremely random and distractive jumble of topics.

2.2.1.3 Automatic Linking

Automatic invocation linking refers to the technique where the terms at the source and link target are both automatically determined by the system. This is the approach that we advocate in order to build the semantic network with minimal manual effort [4, 9].

Our primary viewpoint is that the manual and semi-automatic approaches are an unnecessary burden on contributors, since the knowledge management environment (or Wiki) should contain the data for which concepts are present and how they should be cited. By contrast, authors will usually not be aware of all concepts which are already present within the system – especially for large or distributed corpora.

In addition, a more challenging problem with the manual and semi-automatic linking strategy is that a growing, dynamic corpus will generally necessitate links from existing entries to new entries as the collection becomes larger. To attend to this reality would require continuous re-inspection of the entire corpus by writers or other maintainers, which is an $\mathcal{O}(n^2)$-scale problem (where the corpus contains n entries). To keep an evolving corpus correctly and completely linked, it would be necessary for maintainers to search it upon each update (or at least periodically) to determine if the links in the constituent articles should be updated. When generalizing to inter-linkage across separate corpora, the task would potentially be even more laborious, as authors would have to search across multiple web sites to determine what new terms are available for linking into their entries.

The optimal end product of an automatic invocation linking system should be a fully connected network of articles that will enable readers to navigate and learn from the corpus almost as naturally as if was interlinked by painstaking manual effort. Without understanding the invoked concepts in a statement, the reader cannot attain a complete understanding of the statement, and by extension the entry it appears in. This is why node interlinkage is so important in hypertexts being used as knowledge bases, and why an automated system is of such utility. There are two feasible approaches to automatic linking including rule-based systems and machine learning-based systems. The main focus of this chapter is on rule-based systems but the next section includes a brief introduction to the latest machine learning-based approaches.

2.2.2 Related Technologies

There are a number of technologies that are related or applicable to the *automatic invocation linking* problem. We briefly review them below and discuss their implications and relations to our problem.

2.2.2.1 Semantic Knowledge Bases

There are several efforts [10, 13, 14] towards using a wiki for collaboratively editing semantic knowledge bases where users can specify semantic information including links in addition to standard wiki text. Most of them focus on improving usability and integrating machine readable data and human-readable editable text. PowerMagpie [5] is a tool that was developed that extends browsing by automatically selecting a wide range of online ontologies for a term in a web page that allows users to browse through the ontology and through the entities of the ontology. The system will automatically determine the correct ontology for a term and allow the user to browse that ontology using a browser plugin.

Among the semantic information, links are arguably the most basic and also most relevant markup within a wiki and are interpreted as semantic relations between two concepts described within articles. Völkel et al. [10] provide an extension to be integrated in Wikipedia, that allows users to specify typed links in addition to regular links between articles and typed data inside the articles.

2.2.2.2 Information Retrieval and Web Search

In our automatic linking problem, both the link target and the link source need to be identified and linked automatically. One part of this problem for identifying the best linking target for a concept label bears similarity to the web search problem in finding the most relevant documents based on a keyword. For the most part the work in information retrieval [3] has not been explored in the collaborative semantic linking context [8]. Typical information retrieval issues such as plurality, homonyms, and polysemy are all relevant for the linking process. Some of the information retrieval and web search techniques also provide potential solutions for the linking problem. In particular, the term-frequency and inverse document-frequency (TFIDF) based document ranking may be applied to rank relevant linking targets given a concept label. However, the entries that define a particular concept may not contain the actual concept label (terms) and thus the TFIDF-based approach alone may not yield a good linking quality.

2.2.2.3 Recommender Systems

Another related technology is recommender systems [1] that aim to predict ratings of a particular item for a particular user using a set of similar users based on a user-item rating matrix. At an initial glance, we can model our problem as an entry–entry link matrix where each cell represent a link or non-link from a certain entry to another entry and use entry similarities to help determine the best entry to link to for a term that belongs to a certain entry. While this approach is more appropriate for relevance linking and may help to narrow down the potential link targets, it alone is not sufficient for the invocation or concept linking problem. Nevertheless, it remains an interesting research question to adapt the collaborative filtering technologies to enhance the linking precision by incorporating entry similarities and user feedback into the linking process.

2.2.2.4 Machine Learning

The popularity of Wikipedia has recently produced an interest in the machine learning community for the problem of automatic linking. Wikipedia is a very large data source with hyperlinks manually created by the authors of the wiki. The links in Wikipedia are highly accurate [15]. We can use the existing manually linked pages as a training set for machine learning based automatic linking. The most successful machine learning based technique for automatically linking Wikipedia is described in [11]. We briefly summarize their work. Two different classifiers can be trained for disambiguation and link detection. In the disambiguation phase they use the commonness of each candidate sense and the relatedness to the surrounding context. The commonness of a sense is the probability it is used as a the link destination in Wikipedia. The relatedness or semantic similarity of two pages is based on comparing their incoming and outgoing links. In the detection phase the link detector is trained based on the link probability, the disambiguation confidence, the depth of the article in the Wikipedia classification tree, and the location and spread of the topics mentioned in the page. After using both of the classifiers links can be added to the appropriate location in an entry.

2.3 Challenges and Design Goals

In this section, we discuss the computing challenges and identify a set of design goals for building an automatic invocation linking system.

2.3.1 Linking Quality

The main analytic challenges lie in how to determine which terms or phrases to link and which entries to link to. Typical information retrieval and natural language processing issues such as plurality, homonyms, and polysemy are all relevant for the linking process and bear on the quality of linking. In light of all these challenges, the linking process is necessarily imperfect and so *linking errors* may be present. We characterize many such forms of errors as follows.

- *Mislinking* refers to the error that a term or phrase is linked to an incorrect link target, e.g., an incorrect homonym from a group of homonyms. For example, in our sample entry shown in Fig. 2.1, if “graph” is linked to object 6 instead of 5, then we have a mislink.

- *Overlinking* refers to the error that a term or phrase is linked when there should be no link at all. Note that overlinking also contributes to mislinking because the term is mislinked. For example, if the term "even" is used as a common term (not in a mathematics sense) but was linked to an entry that defines "even number", we have an overlink.
- *Underlinking* refers to the error that a term or phrase is not linked when there should be a link because it invokes a concept that is defined in the corpus. For example, consider our sample entry shown in Fig. 2.1 again, if "planar graph" is not linked, then we have a underlink.

An important goal of designing the automatic linking system is to reduce the above errors and improve the *link precision* (perfect link precision means every link is linked to the correct link target) while maintaining high *link recall* (perfect link recall means a link is created for every concept label that should be linked given the present state of the corpus).

2.3.2 Efficiency and Scalability

Another important design goal of an automatic linking system is its efficiency so the links can be created near-real time during rendering of the entries and its scalability so it can handle the large size of an online encyclopedia corpora. In addition, most collaborative corpora change frequently, an automatic invocation linking system needs to efficiently update the links between entries that are related to newly defined or modified concepts in the corpus. A continually changing corpus must be dealt in such a way that the analysis and processing of automatic links is tractable and scalable.

2.3.3 Generalization to Multiple Corpora

It is also necessary and important that an automatic linking system is easy to use for the adoption by a large user base and easy to setup for the widespread adoption for linking various materials across multiple sites.

To help users learn more quickly it is now generally accepted that knowledge bases should leverage each others' content (or metadata) to increase the scope of the available learning materials. This is the reason for the development of Semantic Web standards such as the Web Ontology Language (OWL). An important design goal of an automatic linking system should be to leverage these standards so that the system would not only enable intra-linking collaborative encyclopedias, such as PlanetMath.org, but also allow for linking educational materials such as lecture notes, blogs, abstracts in research and educational digital libraries. Such usage aids researchers and students in the better understanding of abstracts and full texts, and also helps them find related articles quickly.

2.4 NNexus Approach

We designed and developed NNexus (Noosphere Networked Entry eXtension and Unification System) [4], a system used to automate the process of automatically linking encyclopedia entries (or other definitional knowledge bases) into a semantic network of concepts using metadata of the entries. NNexus is an abstraction and generalization of the automatic linking component of the Noosphere system [9], which is the platform of PlanetMath (planetmath.org), PlanetPhysics (planetphysics.org), and other Noosphere sites. To the best of our knowledge, it is the first automatic linking system that links articles and concepts using the metadata of entries, to make linking almost a "non-issue" for writers, and completely transparent to readers.

NNexus has a number of key features addressing the challenges we outlined above. First, it provides an effective indexing and linking scheme that utilizes metadata to automatically identify link sources and link targets. It achieves perfect link recall without *underlinking* error. It uses a classification-based link steering approach to address the mislinking problem and enhance the link precision. It also provides an interactive entry filtering component to address overlinking problem and further enhance the link precision for a minority of "tough cases." Second, NNexus achieves good efficiency and scalability by its efficient data structures and algorithm design. It has mechanisms for efficiently updating the links between entries that are related to newly defined or modified concepts in the corpus. Lastly, NNexus utilizes OWL and has a simple interface, which allows for an almost unlimited number of online corpora to interconnect for automatic linking.

In this section, we first give an overview of the model and functionalities behind NNexus, then present its key components and techniques.

2.4.1 Overview

Users of NNexus apply the following basic functionality to their corpus: when an entry is rendered either at display time or during offline batch processing, the text is scanned for words or concept labels (*link source*) and they are ultimately turned into hyperlinks to the corresponding entries (*link target*) in the output rendering.

There are two basic steps in performing the invocation linking. The engine breaks the text of an entry into a single words/tokens array to iterate through. The tokens and token tuples (phrases) that invoke concepts defined in other entries are then used for *link target identification* to determine the entries to link to.

Figure 2.2 illustrates the conceptual flow of the automatic linking process. In order to determine which entry to link to for a concept label, NNexus indexes the entries by building a *concept map* that maps all of the concept labels in the corpus to the entries which define these concepts. The tokens and token tuples (phrases) that are identified as link sources are searched to retrieve the candidate links using the concept map (see Section 2.4.2). After the candidate links are determined they are

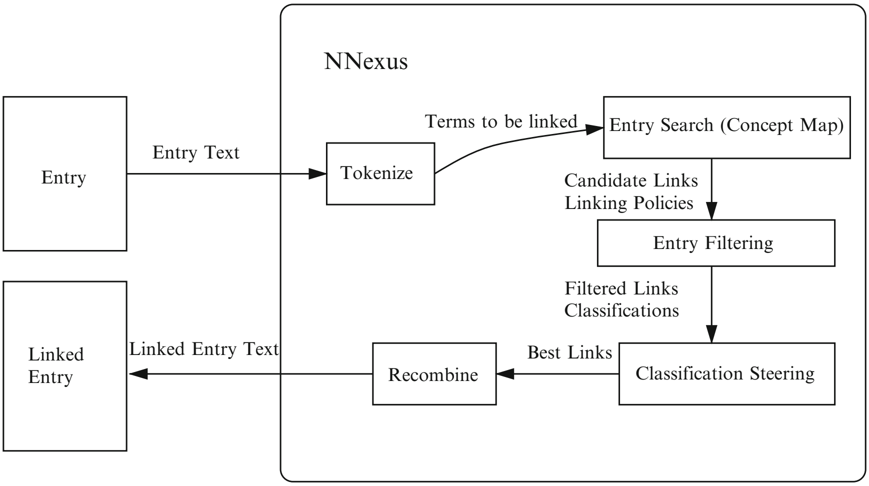


Fig. 2.2 Linking Diagram: When an entry is linked through NNexus the candidate links are found in the concept map. These candidates are then compared against the linking policies and sent through the classification module. The top candidate links are then recombined into the original text and returned to the user

filtered based on linking policies (see Section 2.4.4). The candidates are then compared by "classification proximity" and the object with the closest classification is then selected as the link target (see Section 2.4.3). The "winning" candidate for each position are then substituted into the original text and the linked document is then returned.

In addition, when new concepts are added to the collection (or the set of concept labels otherwise changes), entries containing potential invocation of these concept labels can be *invalidated*. This allows entries to be re-scanned for links, either at invalidation time or before the next time they are displayed. NNexus uses a special structure called the *invalidation index* to facilitate this (see Section 2.4.5).

This automatic system almost completely frees content authors from having to "think about links." It addresses the problems of both outgoing and incoming links, with respect to a new entry or new concepts. However, it is not completely infallible, and in an epistemological sense, there is only so much that a system can infer without having a human-level understanding of the content. Because of this, the user can ultimately override the automatic linking, create their own manual links, or specify link policies for steering the automatic linker (see Section 2.4.4). While complemented and enhanced by the interactive learning components, NNexus is a completely automatic system and we show in next section that NNexus performs well even without any human efforts.

2.4.2 Entry Search

In order to determine which entry to link to for a concept label, NNexus indexes the entries by building a *concept map* that maps all of the concept labels in the corpus to the entries which define these concepts. Below we present the details of how to build the concept map and how it is used for entry search.

When adding a new object (entry) to NNexus, a list of terms the object defines, synonyms, and a title are provided (the concept labels) by the author as metadata. The concept labels are kept in a chained-hash index structure, called the *concept map*. This structure contains as keys the words that occur as the first word of some concept label. Following these words (retrieving the value for the key) leads to a list of full concept labels starting with that particular word. To facilitate efficient scanning of entry text to find concept labels, the map is structured as a chained hash, keyed by the first word of each phrase placed in it. This structure is shown graphically in Fig. 2.3.

NNexus also performs *morphological transformations* on concept labels when building concept map in order to handle morphological invariances and ensure they can be linked to in most typical usages. The first, and most important transformation, has the effect of invariance of pluralization. The second invariance is due to possessiveness. Another morphological invariance concerns international characters. When a token is checked into the index, NNexus will ensure that the token is singular and non-possessive, with a canonicalized encoding.

We now discuss how the concept map is used for entry search. When searching for candidate links for a given entry, the entry is represented as an array of word tokens (concept labels). The tokenized text of the entry is iterated over and searched in the concept map. If a word matches the first word of an indexed concept label in the concept map, the following words in the text are checked to see if they match the longest concept label starting with that word. If this fails, the next longest concept

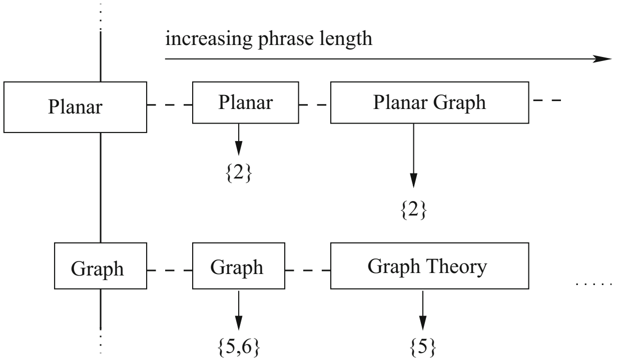


Fig. 2.3 *Concept map:* a fast-access (chained-hash-based) structure filled with all the concept labels for all included corpora, used for determining available linking targets as the text is being scanned. This figure contains a subset that would be generated based on our example corpus

label is checked, and so on. NNexus always performs *longest phrase match*. For example, if an author mentions the phrase “orthogonal function” in their entry and links against a collection defining all of “orthogonal,” “function,” and “orthogonal function,” then NNexus links to the latter. This is based on a nearly universally consistent assumption of natural language, which is that longer phrases semantically subsume their shorter atoms.

When a matching concept label is found, it is included in the *match array*. In our example “graph”, “plane”, and “connected components” are all defined in the corpus. All possible link targets of the terms or phrases are added to the match array. The match array is then iterated over and the possible link targets are then disambiguated to determine the best link target for each term or phrase. Classification based link steering is the main technique used in disambiguation and is discussed in the next section.

2.4.3 Classification Steering

As we discussed in Section 2.3, one of the main challenges of building an automatic linking system is to cope with possible mislinking errors. Online encyclopedias are typically organized into a classification hierarchy, and this ontological knowledge can be utilized to increase the precision of automatic linking by helping identifying the best link targets that are closely related to the link source in the ontological hierarchy. Below we present our classification steering approach that is designed to reduce mislinking errors and to enhance link precision.

2.4.3.1 Classification Hierarchy

Each object in the NNexus corpus may contain one or more classifications. The classification table maps entries (by object ID) to lists of classifications which have been assigned to them by users. The classification hierarchy is represented as a tree. A subtree of the Mathematical Subject Classification (MSC) hierarchy is shown as an example in Fig. 2.4. Each class is represented as a node in the tree. Edges represent parent/child relationships between the classes. In order to select the most relevant link target for a link source, NNexus compares the classes of the candidate link targets to the classes of the link source and selects the closest object with the shortest *distance* in the classification tree. Algorithm 2.1 presents a sketch of the classification steering algorithm.

2.4.3.2 Distance Computation

The key to the algorithm is how to compute the distance between two classes (nodes) in the classification tree. Note that when there are multiple classes associated with

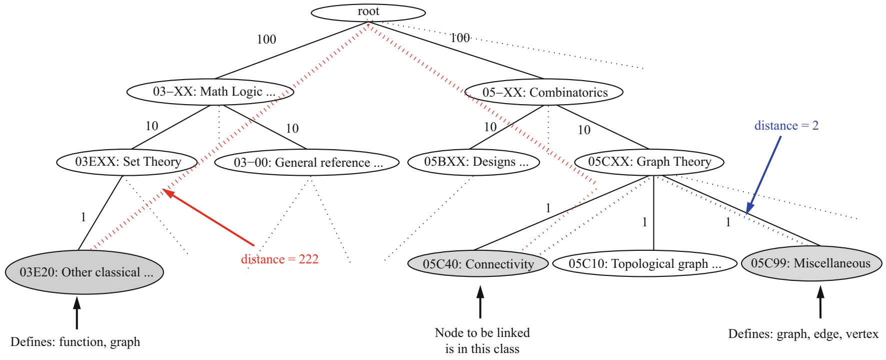


Fig. 2.4 Example Classification Tree: This is the MSC subject classification represented as a weighted graph. The shaded nodes indicates the classification of the source node (where "graph" is to be linked") and the classifications of the two target nodes. The weights are assigned with base 10

Algorithm 2.1 Algorithm of classification steering: it returns the target objects that are closest in classification to the link source in the NNexus classification graph

```
$sourceclasses \leftarrow$ list of classes of source object
$targetobjects \leftarrow$ list of candidate target objects
for all $object_i \in targetobjects$ do
    $targetclasses_i \leftarrow$ list of classes of $object_i$
    $distance_i \leftarrow$ minimum distance between all $sourceclasses$ and $targetclasses_i$ pairs
end for
return $\{object_i | distance_i = \min_i distance_i\}$
```

the link source or link target, the minimum distance of all possible pairs of classes are used. We adopt two approaches, namely *non-weighted approach* and *weighted approach*, for computing the distance between two classes and discuss each them below.

In the *non-weighted approach*, the distance between two classes are simply the length of the shortest path between two classes. Intuitively, a node further away is less related to a given node in the tree. NNexus uses Johnson's All Pairs Shortest Path algorithm to compute the distances between all classes at startup.

In the *weighted approach*, each edge is assigned a weight. This is motivated by the observation that classes at the same level and in the same subtree should be considered closer than classes at a higher level in the same subtree and classes deeper in a subtree are more closely related than classes higher in the same subtree. For example, in Fig. 2.4, 05C10 (Connectivity) and 05C40 (Topological graph ...) are more closely related than the node 05CXX (Graph theory) and 05BXX (Designs ...). Based on this observation, we assign a weight to each edge that is inversely proportional to their depth in the tree. We define a weight of an edge in the graph as

$$w(e) = b^{height-i-1}$$

where b is the chosen base weight (default is 10), *height* is the height of the tree (or in general the distance of the longest path from the designated root node), and i is the distance of the edge from the root. The distance is computed as the weighted shortest path between two nodes. Please note that when the base weight is 1, it becomes the non-weighted approach.

Figure 2.4 also illustrates a scenario of the classification steering algorithm and the distance computation using our example in Fig. 2.1. The MSC classification of our source entry 05C40. The term to be linked, "graph", has two possible link targets: objects 5 with classification 05C99 and object 6 with classification 03E20. We examine the weighted distance (with weight base 10) between the source class and the two target classes to determine which is a better link target. As the weighted distance from 05C99 to 05C40 is shorter in the weighted classification graph than 03E20, "graph" is linked to object 5.

It is worth mentioning that this methodology presents problems when attempting to link across multiple sites (or *across domains*), as different knowledge bases may not use the same classification hierarchy. To address the general problem of inter-linking multiple corpora it is necessary to consider mapping (or otherwise combining) multiple, differing classification ontologies. We are currently investigating the techniques discussed in [2, 12] and implementing this type of functionality in our system.[5]

2.4.4 Entry Filtering

NNexus achieves perfect link recall without underlinking errors as every linkable terms will be linked in an entry. However, it is possible to have overlinking errors when a term that should not be linked (at all) is linked to an entry in the corpus (recall Section 2.3). For example, many articles will contain the word "even." In many cases this is not used in mathematical context and should be forbidden from linking to the entry defining "even number."

In order to combat this overlinking problem and those rare cases where the classification of target articles does not completely disambiguate the link targets, NNexus includes an interactive learning component, entry filtering by linking policies, that is designed to complement and further enhance the link precision by allowing users to specify linking policies. *Linking policies* are a set of directives controlling linking based on the subject classification system within the encyclopedia. The linking policy of an article describes, in terms of subject classes, where links may be made or prohibited. Thus, the entry for "even number" would forbid all articles from linking to the concept "even" unless they were in the number theory category. The author need only supply a linking policy for those terms that the article defines that are used commonly in language and are not meant in a mathematical sense.

[5] For more information on ontology mapping, we recommend the survey in [6].

For each object there is stored text chunk representing the user-supplied linking policy. The linking policy table is keyed by object ID. The linking policies can be specified by the author but administrators also have the ability to modify the linking policies.

We note that the linking policy component requires minimal work from the users and we will show in the experiment section later that by adding linking policies for a very small number (percentage) of entries, the precision for the overall corpus is enhanced significantly.

We are also exploring automatic keyword extraction techniques in order to extract those terms that should be or should not be linked in an automatic way. In addition, we also have a few efforts in progress exploring various ranking techniques by integrating multiple factors such as domain class, priority, pedagogical level, and reputation of the entries to handle the over-linking problem in a more automatic way.

2.4.5 Invalidation

Since NNexus operates on a dynamic and growing corpus we need to know when articles need to be re-linked. As an optimization technique to further enhance the efficiency and performance of the system, NNexus also includes an invalidation component. When a new object is added, NNexus utilizes an *invalidation index* to determine which articles may possibly link to the new object and need to be "invalidated" (marked for re-processing before being displayed again). The invalidation index stores term and phrase *content* information for all entries in the corpus. It is an adaptive index in that longer phrases are only stored if they appear frequently in the collection. There is no limit to how long a stored phrase can be; however, very long phrases are extremely unlikely to appear.

The invalidation index is a variation on a standard text document inverted index structure and works in the usual way for lookups. However, instead of just being keyed on single-word terms, it is keyed on phrases (which are usually but not always single-word). For each term or phrase in the index, there is a list of objects which contain that term or phrase. These lists are called *postings lists*. Since the falloff in occurrence count by phrase length in a typical collection follows a Zipf distribution, the invalidation index tends to be around twice the size of a simple word-based inverted index.

The invalidation index has a special property that for every phrase indexed, all shorter prefixes of that phrase are also indexed for every occurrence of the longer phrase. This allows us to guarantee that occurrences of the shorter phrases or single terms will be noticed if we do a lookup using these shorter tuples as keys.

The invalidation index exists for a single purpose: so that when concept labels are added to the collection (or when they change), we can determine a minimal superset of entries effected by the change – that is, they likely link to the newly added concept. The invalidation index allows us to do this in a way that never misses an entry that should be re-examined, but does not catch too many irrelevant entries (false positives).

2.5 Case Studies

We have implemented NNexus as a general, open source tool and deployed NNexus in a variety of settings including the online encyclopedia web site PlanetMath.org. In this section, we briefly introduce some implementation features and the interface of NNexus, present some statistics of its deployment in the PlanetMath corpus, and discuss a few other scenarios illustrating the deployability and effectiveness of NNexus. Figure 2.5 shows a sample architecture linking to multiple corpora.

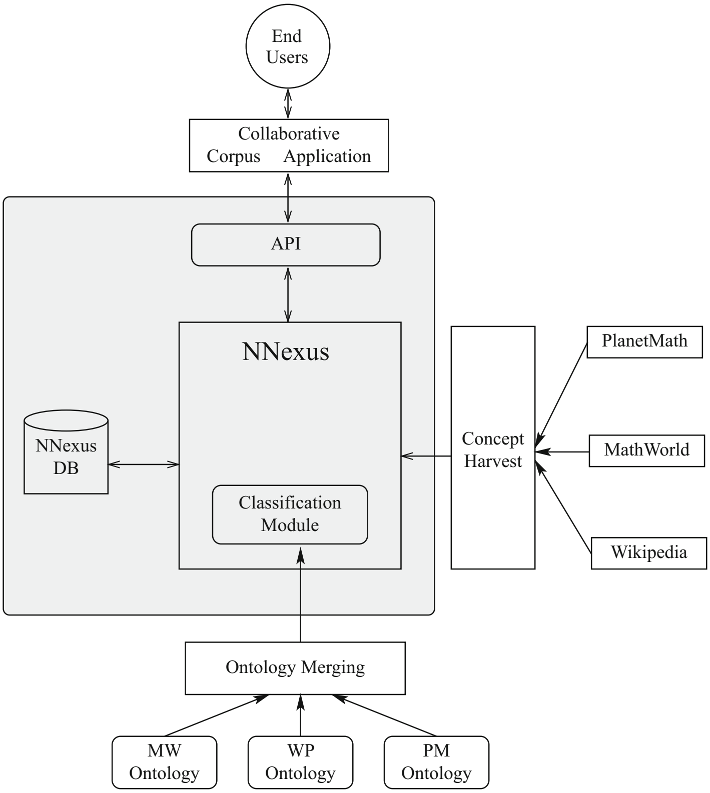


Fig. 2.5 NNexus System Architecture (in an example deployment): The shaded region denotes NNexus proper. The classification module provides classification-invariant link steering between multiple ontologies

2.5.1 PlanetMath

The core methods of NNexus have proven their large-scale applicability in the PlanetMath[6] site, a collaborative and dynamic mathematics encyclopedia in existence on the web for about 7 years now. As of this writing PlanetMath had more than 7,145 entries defining more than 12,171 concepts. We present a set of statistics of the linking system for the PlanteMath Corpus in terms of linking quality, efficiency and scalability.

2.5.1.1 Linking Quality

In order to evaluate the linking quality and the effect of the different system components of NNexus, we performed a study on the live PlanetMath collection examining the linking precision with basic lexical matching (without classification steering and linking policies) vs. lexical matching with classification steering (without linking policies) vs. lexical matching with classification steering and linking policies. The experiment was performed over the entire PlanetMath corpus but the statistics were estimated from a sample of 50 random entries in the corpus.

Table 2.1 presents the link precision for the three cases respectively. Link *precision* is defined as the number of *correct* links (those to the appropriate destination) divided by the number of created links. Note that NNexus system was designed for near-perfect link *recall* defined as the number of created (retrieved) links divided by the number of concepts invoked the entry that are actually defined in the corpus and thus we do not report link recall. We observe that the classification steering as well as linking policies improve the link precision significantly. Note that these policies were supplied by real-world users with no prompting, and no effort was made to tackle the remaining problematic cases of overlinking. Nevertheless, the linking policies drove precision up to more than 92%.[7]

Table 2.1 Linking quality for PlanetMath

Statistic	Basic	Steering	Steering and policies
Number of links	761	761	701
Good links	630	672	646
Mislinks	131	89	55
Overlinks	69	69	36
% mislink	17.2	11.7	7.8
% overlink	9.1	9.1	5.1
Precision	82.8	88.3	92.2

[6] Visit PlanetMath on the web at http://www.planetmath.org

[7] Likely this number could exceed 95% with a little bit of targeted effort, and given that these policies have been available on PlanetMath for less than 2 years, the numbers will likely continue to improve on their own.

We believe these results provide compelling support for our hypothesis that NNexus with classification-based link steering achieves good linking quality. Further, overlinking, which represents at least two-thirds of the precision shortfall in our collection, can be largely eliminated by adding linking policies to a small subset of it. The results also indicate that by adding the linking polices the mislinking percentage was reduced. Thus a small subset of homonyms in the corpus contribute not only to overlinking by also to much of the mislinking.

2.5.1.2 Scalability and Efficiency

To study the scalability and efficiency of our approach, we ran experiments on a modest Mac machine running OS X with a 1.83 GHz Intel Core Duo and 512MB DDR2 SDRAM. We selected random subsets of size 200–7,132 from the PlanetMath corpus and kept track of the number of seconds to link every object in the subset corpora.

Table 2.2 and Fig. 2.6 show the performance results for different corpus sizes. We can see that the time per link quickly falls off and then hovers around a constant value as the collection grows. This indicates that NNexus is not only efficient but also scalable to very large corpus sizes. All overhead quickly amortizes and diminishes relative to productive linking work done by the system, meaning that NNexus automatic linking is a legitimate feature to build into expanding collections and growing ensembles of interlinked collections on the web.

2.5.1.3 Comparison to Wikipedia

A survey in [15] shows that about 97–99% of Wikipedia links are accurate. However, this study is not directly comparable to our survey for a number of reasons. First, because it relies on the convention of "disambiguation nodes" (which NNexus allows one to avoid) and second, it doesn't take into account link recall (underlinking). In other words, links in Wikipedia tend to be accurate, but some of

Table 2.2 Linking scalability for PlanetMath

Corpus size	Number of links	Total time (s)	Time/link
200	640	126	0.197
500	2,067	290	0.140
1,000	5,837	617	0.106
2,000	17,757	1,218	0.069
3,000	35,682	1,972	0.055
4,000	52,030	2,881	0.055
5,000	79,139	3,737	0.047
6,000	101,787	4,487	0.044
7,132	127,430	5,599	0.044

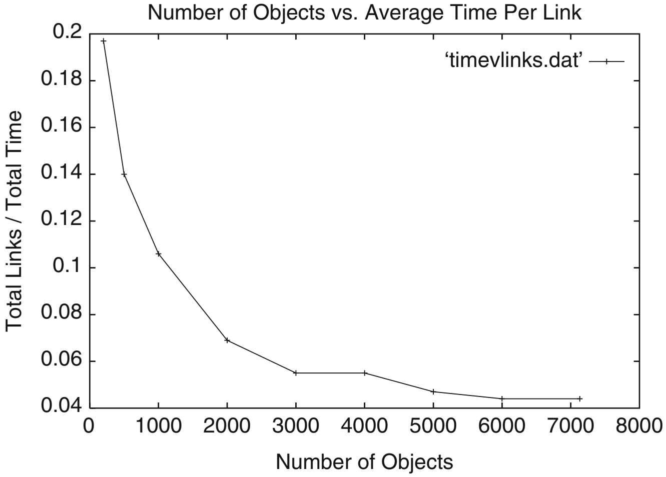


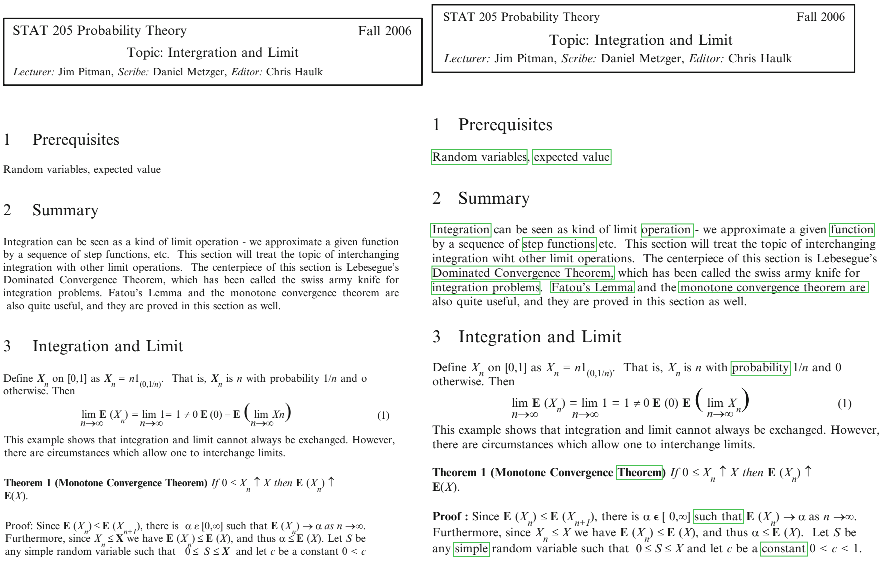
Fig. 2.6 Scalability study: time-per-link for progressively larger corpora, showing clearly that the automatic linking process is sub-linear in time complexity

this "accuracy" is due to the presence of disambiguation nodes, and some is likely due to the fact that many links simply aren't being made.

Most significantly, from a usability and productivity standpoint, no formal comparison of the *effort* required for link maintenance in the manual/semi-automatic vs. automatic paradigms has been made. However, anecdotal evidence suggests our approach to linking is less work for authors and more appreciated by them; and with classification and linking steering, precision approaches that achieved on Wikipedia with manual effort and disambiguation nodes. It is interesting to note that artificial hubs are created in the Wikipedia network because of disambiguation pages. This may have impact on some algorithms that use the link structure of a semantic network such as HITS [7]. Disambiguation pages paradoxically add ambiguity to the data because the link structure is modified and it encourages authors not to find the correct target for a link.

2.5.2 Lecture Notes

In addition to enabling intra-linking in an single encyclopedic knowledge base such as PlanetMath, NNexus also provides a generalized automatic linking solution to a variety of potential applications. One such application is the linking of lecture notes to math encyclopedia sites (including PlanethMath and MathWorld, but potentially extending to others, such as Wikipedia, the Digital Library of Mathematical Functions [DLMF], and more). Figure 2.7 demonstrates this sort of use, showing screenshots of automatically linked notes from a probabilities course taught by



STAT 205 Probability Theory Fall 2006

Topic: Intergration and Limit

Lecturer: Jim Pitman, *Scribe:* Daniel Metzger, *Editor:* Chris Haulk

1 Prerequisites

Random variables, expected value

2 Summary

Integration can be seen as a kind of limit operation - we approximate a given function by a sequence of step functions, etc. This section will treat the topic of interchanging integration with other limit operations. The centerpiece of this section is Lebesegue's Dominated Convergence Theorem, which has been called the swiss army knife for integration problems. Fatou's Lemma and the monotone convergence theorem are also quite useful, and they are proved in this section as well.

3 Integration and Limit

Define $\boldsymbol{X}_n$ on [0,1] as $\boldsymbol{X}_n = n1_{(0,1/n)}$. That is, $\boldsymbol{X}_n$ is n with probability $1/n$ and o otherwise. Then

$$\lim_{n\to\infty} \mathbf{E}\,(X_n) = \lim_{n\to\infty} 1 = 1 \neq 0\ \mathbf{E}\,(0) = \mathbf{E}\left(\lim_{n\to\infty} Xn\right) \tag{1}$$

This example shows that integration and limit cannot always be exchanged. However, there are circumstances which allow one to interchange limits.

Theorem 1 (Monotone Convergence Theorem) *If* $0 \leq X_n \uparrow X$ *then* $\mathbf{E}\,(X_n) \uparrow \mathbf{E}(X)$.

Proof: Since $\mathbf{E}\,(X_n) \leq \mathbf{E}\,(X_{n+1})$, there is $\alpha\ \varepsilon\ [0,\infty]$ such that $\mathbf{E}\,(X_n) \to \alpha$ *as* $n \to \infty$. Furthermore, since $X_n \leq \mathbf{X}$ we have $\mathbf{E}\,(X_n) \leq \mathbf{E}\,(X)$, and thus $\alpha \leq \mathbf{E}\,(X)$. Let S be any simple random variable such that $0 \leq S \leq \boldsymbol{X}$ and let c be a constant $0 < c$

STAT 205 Probability Theory Fall 2006

Topic: Integration and Limit

Lecturer: Jim Pitman, *Scribe:* Daniel Metzger, *Editor:* Chris Haulk

1 Prerequisites

Random variables, expected value

2 Summary

Integration can be seen as kind of limit operation - we approximate a given function by a sequence of step functions etc. This section will treat the topic of interchanging integration wiht other limit operations. The centerpiece of this section is Lebesegue's Dominated Convergence Theorem, which has been called the swiss army knife for integration problems. Fatou's Lemma and the monotone convergence theorem are also quite useful, and they are proved in this section as well.

3 Integration and Limit

Define X_n on [0,1] as $X_n = n1_{(0,1/n)}$. That is, X_n is n with probability $1/n$ and 0 otherwise. Then

$$\lim_{n\to\infty} \mathbf{E}\,(X_n) = \lim_{n\to\infty} 1 = 1 \neq 0\ \mathbf{E}\,(0)\ \mathbf{E}\left(\lim_{n\to\infty} X_n\right) \tag{1}$$

This example shows that integration and limit cannot always be exchanged. However, there are circumstances which allow one to interchange limits.

Theorem 1 (Monotone Convergence Theorem) *If* $0 \leq X_n \uparrow X$ *then* $\mathbf{E}\,(X_n) \uparrow \mathbf{E}(X)$.

Proof : Since $\mathbf{E}\,(X_n) \leq \mathbf{E}\,(X_{n+1})$, there is $\alpha \in [\,0,\infty]$ such that $\mathbf{E}\,(X_n) \to \alpha$ as $n \to \infty$. Furthermore, since $X_n \leq X$ we have $\mathbf{E}\,(X_n) \leq \mathbf{E}\,(X)$, and thus $\alpha \leq \mathbf{E}\,(X)$. Let S be any simple random variable such that $0 \leq S \leq X$ and let c be a constant $0 < c < 1$.

Fig. 2.7 Screenshot of original (*left*) and automatically linked (*right*) lecture notes using NNexus. The links in this example are to definitions on both MathWorld and PlanetMath, depending on which site had each particular definition available, and in the case both did, a collection priority configuration option determined the outcome. Concepts were "imported" from MathWorld using that site's OAI repository

Professor Jim Pitman at UC Berkeley – before and after automatic linking with NNexus (the links in this example are to both PlanetMath and MathWorld).

Due to the ease-of-use and success of linking lecture notes we are confident that we can extend NNexus to other applications with diminishing additional effort. Another interesting application is the linking of abstracts in research and educational digital libraries. This would enable learners (students or researchers) to quickly find related articles and also would help the user better understand the underlying concepts in the abstracts.

It would also be useful to apply automatic linking to educational blogs, which are of increasing prevalence and impact on the web, and are being embraced by large-scale efforts such as the NSDL.[8]

The modular design of NNexus allows developers to use NNexus as a web plugin for on-demand text linking and for various document authoring applications. NNexus could be deployed as a web service to allow third parties to link arbitrary documents to particular corpora.

[8] For their "Expert Voices" service. See http://www.nsdl.org/

2.6 Conclusion and Open Issues

We have introduced the problem of automatic invocation linking for collaborative web-based corpora and outlined the design goals that any automatic linking system should strive to achieve. We presented NNexus, an automatic linking system that we have developed as a potential solution and presented a few case studies demonstrating the effectiveness and efficiency of the NNexus approach. NNexus is now available for general use as open source software[9] and we look forward to working with others to improve it and apply it more widely to enhance the semantic quality of the web in general.

There are a number of open research directions remaining with the automatic linking problem. First, in order to achieve perfect link recall yet avoiding overlinking problem, automatic keyword extraction technique is a promising direction to investigate to better extract concept labels to be linked. Second, in order to further enhance link precision, it is a fruitful research direction to combine content based information retrieval techniques and collaborative filtering techniques [1] with the metadata based approach in NNexus to enhance the ranking of potential link targets and address issues of "competing" entries and different needs and preferences of authors. This especially becomes an issue when one goes beyond a single collaborative corpus, as would typically be the case in linking to them by third parties. Finally, it remains a major research and development item to generalize any linking system for inter-linking of multiple corpus across domains with expansion of ontology mapping capabilities.

Acknowledgements This work has been partially supported by the Google Summer of Code Program. We would also like to thank the editors of the special issue and the anonymous reviewers for their valuable comments that improved this paper.

References

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 2005.
2. Zharko Aleksovski and Michel Klein. Ontology mapping using background knowledge. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, 2005.
3. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
4. J. Gardner, A. Krowne, and L. Xiong. NNexus: An Automatic Linker for Collaborative Web-Based Corpora In *IEEE Transactions on Knowledge and Data Engineering*, 21(6), 2009.
5. L. Gridinoc, M. Sabou, M. d́Aquin, M. Dzbor, and E. Motta Semantic Browsing with Power-Magpie In *ESWC '2008: 5th European Semantic Web Conference*, pages 802–806, 2008.

[9] http://aux.planetmath.org/nnexus/

6. Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: The state of the art. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.
7. Kleinberg, Jon *Authoritative sources in a hyperlinked environment* In *Journal of the ACM 46* (5):604632. 1999.
8. J. Kolbitsch and H. Maurer. Community building around encyclopeadic knowledge. *Journal of Computing and Information Technology*, 14, 2006.
9. Aaron Krowne. An architecture for collaborative math and science digital libraries. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2003.
10. Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 585–594, New York, NY, USA, 2006. ACM Press.
11. D. Milne and I. Witten. Learning to Link with Wikipedia. In *CIKM '2008: 17th Conference on Information and Knowledge Management*, 2008.
12. Natalya Fridman Noy and Mark A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000.
13. S. E. Roberto Tazzoli and Paolo Castagna. Towards a semantic wiki web. In *In Demo Session at ISWC2004*, 2004.
14. Adam Souzis. Building a semantic wiki. *IEEE Intelligent Systems*, 20(5):87–91, 2005.
15. G. Weaver, B. Strickland, and G. Crane. Quantifying the accuracy of relational statements in wikipedia: a methodology. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, 2006.