

NNexus: An Automatic Linker for Collaborative Web-Based Corpora

James Gardner
Department of Math/CS
Emory University
jgardn3@emory.edu

Aaron Krowne
PlanetMath.org
akrowne@gmail.com

Li Xiong
Department of Math/CS
Emory University
lxiong@mathcs.emory.edu

ABSTRACT

Collaborative online encyclopedias or knowledge bases such as Wikipedia and PlanetMath are becoming increasingly popular. We demonstrate NNexus, a generalization of the automatic linking engine of PlanetMath.org and the first system that automates the process of linking disparate “encyclopedia” entries into a fully-connected conceptual network. The main challenges of this problem space include: 1) linking quality (correctly identifying which terms to link and which entry to link to with minimal effort on the part of users), 2) efficiency and scalability, and 3) generalization to multiple knowledge bases and web-based information environment. We present NNexus that utilizes subject classification and other metadata to address these challenges and demonstrate its effectiveness and efficiency through multiple real world corpora.

1. INTRODUCTION

Collaborative online encyclopedias or knowledge bases such as Wikipedia¹ and PlanetMath² 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, it is critical that users of any online reference are able to easily “jump” to requisite concepts in the network in order to fully understand the current one. For full comprehension, these jumps should extend all the way “down” to 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. 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.

Problem definition. 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. For our purpose, 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*. Concept invocation refers to the mention of a concept in the text. Any statement in a language is composed of concepts represented by tuples of words. We call the tuples of words representing a concept *concept label*. An *invocation link* is a hyperlink from a concept label to an entry that defines the concept. We refer to the concept label being linked from as *link source* and the entry being linked to as *link target*. The problem of *invocation linking* is how to add invocation links in a collaborative online encyclopedia.

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
...

Table 1: Example Document Corpora

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 ...

Figure 1: Example Entry Text

Table 1 shows a list of entries (objects) in an example online encyclopedia³ corpus with their object ID and metadata

³<http://planetmath.org>

¹<http://www.wikipedia.org>
²<http://www.planetmath.org>

including what concepts each entry defines and the Mathematical Subject Classification (MSC) for each entry. Figure 1 shows an example entry⁴ with links to concepts that are defined in the same corpus. The terms underlined indicate terms (concept labels) that need to be linked based on the meta-data in the table. For example, *planar graph* in the example entry needs to be linked to object (entry) 2 that defines the concept *planar graph*. We will use this example to explain the concepts discussed in this paper.

Existing and potential solutions. The existing linking approaches can be mainly classified into *manual linking* where both the link source and link target are explicitly defined (such as blog software) and *semi-automatic linking* where the link source are explicitly marked but the link target is determined by the collaborative online encyclopedia system (such as current online encyclopedias including Wikipedia). There are several efforts that involve collaboratively editing semantic knowledge bases where users specify the semantic information including links in addition to the standard wiki text [8, 7, 9]. The perspective taken in our work is that the manual and semi-automatic approaches are an unnecessary burden on contributors. In addition, with the manual and semi-automatic linking strategy, a growing and dynamic corpus will require continuous re-inspection of the entire corpus by writers or other maintainers.

Part of the linking problem in identifying the best linking target bears similarities to the search problem on the web. However, for the most part most of the work in information retrieval [1] has not been explored in the linking context [5], which is unsurprising given the novelty of collaboratively-built knowledge-bases. In addition, in our problem not only the link target but also the link source need to be identified and linked automatically. The INEX Link-The-Wiki Track [4] was recently created because the automatic linking problem is now an interest in the XML and semantic web forum. The closest work to ours is that of Milne and Witten [6]. They use a machine learning approach to the linking problem, by training their automatic linking system on input from wikipedia authors as ground truth. We are currently investigating similar techniques by using different machine learning techniques to enhance our automatic linking system to more general problems, such as linking Wikipedia or other knowledge bases that do not have a rich or manageable classification hierarchy.

Contributions. We designed and developed NNexus⁵ (Nosphere Networked Entry eXtension and Unification System) [2, 3], a system used to automatically link encyclopedia entries (or other definitional knowledge bases) into a semantic network of concepts using metadata of the entries.

We summarize the research contributions of NNexus below. First, to the best of our knowledge, NNexus 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. we advocate and formalize the automatic invocation linking problem and identifies the key technical challenges and design goals for building such a system. Second, NNexus includes an effective indexing and linking scheme that utilizes metadata to automatically identify link sources and link targets. It uses a classification-based link steering approach

to enhance the link precision. It also provides an interactive entry filtering component to further enhance the link precision for a minority of “tough cases.” Finally, 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. We performed extensive experimental evaluations using real online corpora demonstrating the feasibility and benefits of using an automatic linking system.

As a software, NNexus was designed and developed to have minimum amount of dependencies and with an API so that it can be used with any document corpus and with client software written in any programming language⁶. It utilizes OWL and has a simple interface, which allows for an almost unlimited number of online corpora to interconnect for automatic linking. Users can use NNexus to link their corpus with their own or other knowledge bases such as Wikipedia or PlanetMath.

2. NNEXUS FRAMEWORK

In this section, we present an overview of NNexus and discuss key techniques and features in its framework.

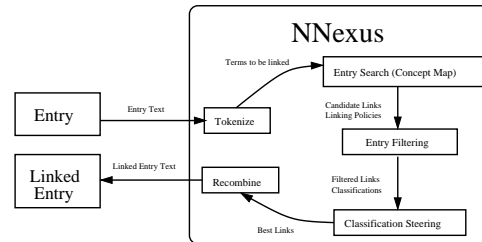


Figure 2: NNexus Linking Diagram

Overview. Figure 2 illustrates the conceptual flow of the automatic linking process in NNexus. 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 order to determine the link targets for a concept label, the *entry search* component searches for corresponding entries using a *concept map* that maps all of the concept labels in the corpus to the entries which define these concepts. The *entry filtering* component filters the candidate links based on linking policies. The best candidate is then selected by the *classification steering* component. In addition, an *invalidation* component is designed to invalidate entries when new concepts are added to the collection (or the set of concept labels otherwise changes). We describe each of the key components below.

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. 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

⁴<http://planetmath.org/encyclopedia/PlaneGraph.html>

⁵NNexus is released under an MIT/X11 style license.

⁶NNexus only requires a database system (currently MySQL is supported) and some Perl XML packages (available from CPAN).

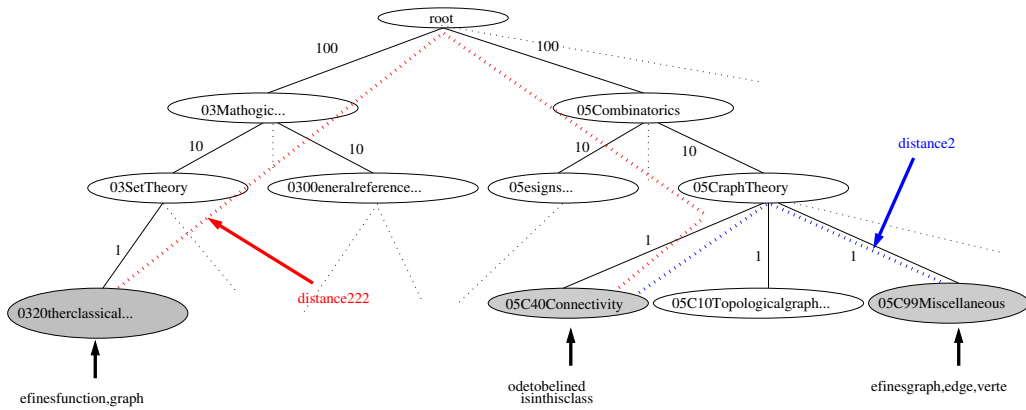


Figure 3: Illustration of Classification Steering

maintained in an index structure, called the *concept map*. 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. 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 (allows for pluralization, possessiveness, and canonicalization).

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 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.

Classification steering. One of the main challenges of building an automatic linking system is to cope with possible mislinking errors where a term or phrase is linked to an incorrect link target. 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.

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 Figure 3. 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.

In our approach each edge in the tree is assigned a weight. This is motivated by the observation that classes at a lower (deeper) level in the same subtree of the hierarchy are more closely related than classes at a higher level in the same subtree. For example, in Figure 3, 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

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

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. NNexus uses Johnson’s All Pairs Shortest Path algorithm to compute the distances between all classes at startup.

Entry filtering. NNexus achieves perfect link recall as every linkable concept 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. 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 are used commonly in language and are not meant in a mathematical sense.

Invalidation. 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.

3. DEMONSTRATION

In the demonstration, we will: (a) show the basic automatic linking functionality of NNexus (user’s view) and its effectiveness and efficiency, (b) show the implementation, installation, configuration and operation of NNexus (administrator’s view), and (c) give an under-the-hood look behind some of the key techniques of NNexus. The demonstration will be highly interactive, allowing users to play with the system, e.g. selecting articles to link and examining the article before and after linking as well as the articles that it links to.

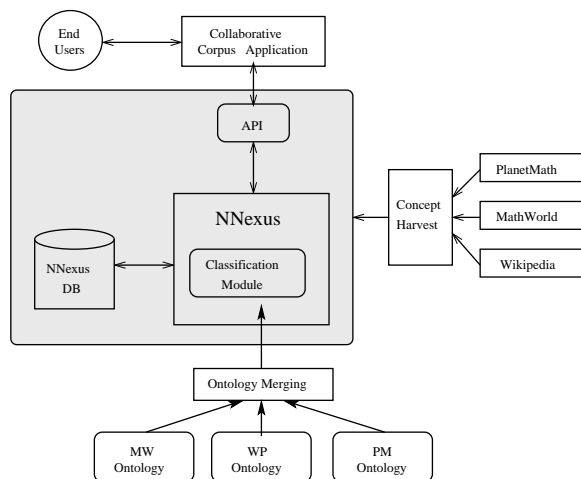


Figure 4: NNexus Deployment Architecture

Setup. Figure 4 shows the planned demonstration deployment diagram of NNexus that links a corpora with multiple knowledge bases. We will use four corpora – PlanetMath (7145 entries declaring more than 12000 concepts), Jim Pitman’s Probability course notes ⁷, MathWorld⁸ and Wikipedia. We will show how to load these corpora into the NNexus database and how NNexus can operate on a single or multiple servers. Domain ontologies are used for enhancing the linking quality (which we will discuss later). If the internet connection is available in the demonstration room, we will show how the corpus will be linked to online entries. We will also have a subset of online entries copied on our local machine in case the internet connection is not available. **User view.** We will show how to load entries into NNexus, including how to specify and modify the classification and

user specified linking policies. This will include how to do this programmatically as well as through a user interface developed for NNexus. We will show various documents before and after linking and demonstrate the linking quality. We will also demonstrate batch processing and show statistics in terms of linking quality and linking efficiency.

Administrator view. We will show how to install, configure, and administer the NNexus system. This includes how administrators can configure NNexus for multiple corpora, how to modify linking policies, and how to configure NNexus to display multiple link targets for different domains.

Under-the-Hood. We will also show NNexus in operation by examining how NNexus makes linking decision through classification steering when there are multiple linking targets.

We will devise a few linking scenarios including some tough cases to show how the classification steering work and how it helps to disambiguate possible link targets. Figure 3 illustrates such a demonstration scenario. We will load the article that defines “planar graph” from PlanetMath⁹ (a snippet is shown in Figure 1). The system identifies that the term “graph” needs to be linked and it has two possible link targets and both of them define “graph” among other things. The first one defines “function”, “graph”, etc while the second one defines “graph” as well as “edge” and “vertex”. If we examine the MSC classification of the target entries, the first one is 05C99 and the second one is 03E20. The MSC classification for our source entry is 05C40. We will demonstrate how NNexus computes the weighted distance 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” will be linked to the second entry which is the article that defines “graph” in PlanetMath¹⁰. We will show the source article as well as both target articles to the demonstration audience so that they can examine them and verify whether the correct or more appropriate link target is selected by NNexus.

4. REFERENCES

- [1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
- [2] J. Gardner, A. Krowne, and L. Xiong. NNexus: Towards a Subject-Steered Automatic Linker for a Massively-Distributed Collaborative Corpus. In *2nd IEEE CollaborateCom Workshop*, 2006.
- [3] J. Gardner, A. Krowne, and L. Xiong. NNexus: An automatic linker for collaborative web-based corpora. *IEEE Transactions on Knowledge and Data Engineering*, 2008.
- [4] D. W. Huang, Y. Xu, A. Trotman, and S. Geva. Overview of INEX 2007 link the wiki track. pages 373–387, 2008.
- [5] J. Kolbitsch and H. Maurer. Community building around encyclopedic knowledge. *Journal of Computing and Information Technology*, 14, 2006.
- [6] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, pages 509–518, New York, NY, USA, 2008. ACM.
- [7] A. Souzis. Building a semantic wiki. *IEEE Intelligent Systems*, 20(5):87–91, 2005.
- [8] S. E. R. Tazzoli and P. Castagna. Towards a semantic wiki wiki web. In *In Demo Session at ISWC2004*, 2004.
- [9] M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. 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.

⁷<http://www.stat.berkeley.edu/~pitman>

⁸<http://www.mathworld.com>

⁹<http://planetmath.org/encyclopedia/PlaneGraph.html>

¹⁰<http://planetmath.org/encyclopedia/Adjacent.html>