Constructing a Focused Taxonomy from a Document Collection

Olena Medelyan,[1] Steve Manion,[1] Jeen Broekstra,[1] Anna Divoli,[1]
Anna-Lan Huang,[2] Ian H. Witten[2]

[1] Pingar Research, Auckland, New Zealand
(alyona.medelyan|steve.manion|anna.divoli)@pingar.com,
jeen@rivuli-development.com
[2] University of Waikato, Hamilton, New Zealand
(ahuang|ihw)@cs.waikato.ac.nz

Abstract. We describe a new method for constructing custom taxonomies from document collections. It involves identifying relevant concepts and entities in text; linking them to knowledge sources like Wikipedia, DBpedia, Freebase, and any supplied taxonomies from related domains; disambiguating conflicting concept mappings; and selecting semantic relations that best group them hierarchically. An RDF model supports interoperability of these steps, and also provides a flexible way of including existing NLP tools and further knowledge sources. From 2000 news articles we construct a custom taxonomy with 10,000 concepts and 12,700 relations, similar in structure to manually created counterparts. Evaluation by 15 human judges shows the precision to be 89% and 90% for concepts and relations respectively; recall was 75% with respect to a manually generated taxonomy for the same domain.

1 Introduction

Domain-specific taxonomies constitute a valuable resource for knowledge-based enterprises: they support searching, browsing, organizing information, and numerous other activities. However, few commercial enterprises possess taxonomies specialized to their line of business. Creating taxonomies manually is laborious, expensive, and unsustainable in dynamic environments (e.g. news). Effective automatic methods would be highly valued.

Automated taxonomy induction has been well researched. Some approaches derive taxonomies from the text itself [1], some from Wikipedia [2], while others combine text, Wikipedia and possibly WordNet to either extend these sources with new terms and relations [3] or carve a taxonomy tailored to a particular collection [4,5]. Our research falls into the last category, but extends it by defining a framework through which any combination of knowledge sources can drive the creation of document-focused taxonomies.

We regard taxonomy construction as a process with five clearly defined stages. The first, initialization, converts documents to text. The second extracts concepts and named entities from text using existing NLP tools. The third connects

named entities to Linked Data sources like Freebase and DBpedia. The fourth identifies conflicting concept mappings and resolves them with an algorithm that disambiguates concepts that have matching labels but different URIs. The fifth connects the concepts into a single taxonomy by carefully selecting semantic relations from the original knowledge sources, choosing only relations that create meaningful hierarchies given the concept distribution in the input documents. These five stages interoperate seamlessly thanks to an RDF model, and the output is a taxonomy expressed in SKOS, a standard RDF format.

The method itself is domain independent—indeed the resulting taxonomy may span multiple domains covered by the document collection and the input knowledge sources. We have generated and made available several such taxonomies from publicly available datasets in five different domains.[3] This paper includes an in-depth evaluation of a taxonomy generated from news articles. Fifteen human judges rated the precision of concepts at 89% and relations at 90%; recall was 75% with respect to a manually built taxonomy for the same domain. Many of the apparently missing concepts are present with different—and arguably more precise—labels.

Our contribution is threefold: (a) an RDF model that allows document-focused taxonomies to be constructed from any combination of knowledge sources; (b) a flexible disambiguation technique for resolving conflicting mappings and finding equivalent concepts from different sources; and (c) a set of heuristics for merging semantic relations from different sources into a single hierarchy. Our evaluation shows that current state-of-the-art concept and entity extraction tools, paired with heuristics for disambiguating and consolidating them, produce taxonomies that are demonstrably comparable to those created by experts.

2 Related Work

Automatic taxonomy induction from text has been studied extensively. Early corpus-based methods extract taxonomic terms and hierarchical relations that focus on the intrinsic characteristics of a given corpus; external knowledge is rarely consulted. For example, hierarchical relations can be extracted based on term distribution statistics [6] or using lexico-syntactic patterns [7,1]. These methods are usually unsupervised, with no prior knowledge about the corpus. However, they typically assume only a single sense per word in the corpus, and produce taxonomies based on words rather than word senses.

Research has been conducted on leveraging knowledge bases to facilitate taxonomy induction from both closed- and open-domain text collections. Some researchers derive structured taxonomies from semi-structured knowledge bases [2,8] or from unstructured content on the Web at large [9]. Others expand knowledge bases with previously unknown terms and relations discovered from large corpora—for example, Matuszek et al. enrich the Cyc knowledge base with information extracted from the Web [10], while Snow et al. expand WordNet with new synsets by using statistical classifiers built from lexical information extracted

[3] `http://bit.ly/f-step`

from news articles [3]. Still others interlink documents and knowledge bases: they match phrases in the former with concepts in the latter [11,12] and identify taxonomic relations between them [4,5]. These studies do address the issue of sense ambiguity: polysemous phrases are resolved to their intended senses while synonyms are mapped to the same concept. However, they typically only consult a single source and users do not intervene in the taxonomy construction process.

The Castanet project [4] and Dakka and Ipeirotis's research [5] relate closely to our work. They both derive hierarchical metadata structures from text collections and both consult external sources—WordNet in the former case and Wikipedia, WordNet and the Web in the latter—to find important concepts in documents. Castanet identifies taxonomic relations based on WordNet's `is-a` relations, whereas Dakka and Ipeirotis use subsumption rules [6]. The latter only select those taxonomic concepts for final groupings that occur frequently in the documents in non-related contexts. In contrast to our work, both studies represent the extracted information as hierarchical faceted metadata: the outcome is no longer a single taxonomy but is instead split into separate facets. Although Dakka and Ipeirotis consult multiple sources, they do not check which concepts are the same and which are different. In contrast, we explicitly address the problem of sense disambiguation and consolidation with multiple sources.

Our work also intersects with research on relation extraction and ontology induction from text, the closest being [13], which also links phrases in text to Wikipedia, DBpedia and WordNet URIs, extracts relations, and represents them as RDF. However, their input is a single short piece of text, whereas we analyze an entire document collection as a whole, and focus on organizing the information hierarchically.

3 Architecture of the Taxonomy Generator

The primary input to our taxonomy generator is a collection of documents and, optionally, a taxonomy for a related domain (e.g., the Agrovoc thesaurus or the Gene ontology). Our system automatically consults external knowledge sources, and links concepts extracted from the documents to terminology in these sources. By default we use Freebase, DBpedia and Wikipedia, but domain-specific linked data sources like Geonames, BBC Music, or the Genbank Entrez Nucleotide database can also be consulted.[4] Finally, a small taxonomy with preferred root nodes can be supplied to guide the upper levels of the generated taxonomy.

3.1 Defining Taxonomies in SKOS

The result of each step of the taxonomy generation process is stored as an RDF data structure, using the Simple Knowledge Organization System vocabulary. SKOS is designed for sharing and linking thesauri, taxonomies, classification schemes and subject heading systems via the Web.[5] An SKOS model consists

[4] Suitable linked data sources can be found at `http://thedatahub.org/group/lodcloud`
[5] See `http://www.w3.org/2004/02/skos`

of a hierarchical collection of *concepts*, defined as "units of thought"—abstract entities representing ideas, objects or events. A concept is modeled as an instance of the class `skos:Concept`. An `skos:prefLabel` attribute records its preferred name and `skos:altLabel` attributes record optional synonyms. Concepts are linked via semantic relations such as `skos:broader` (to indicate that one concept is broader in meaning than another) and its inverse `skos:narrower`. These relations allow concepts to be structured into a taxonomic hierarchy.

Our goal is to produce a new knowledge organization system (a taxonomy) based on heterogeneous sources, including concepts extracted from text as well as concepts in existing sources, and SKOS is a natural modeling format. Also, many existing public knowledge systems are available online as SKOS data,[6] and reusing these sources ensures that any taxonomy we generate is immediately linked via concept mappings to third-party data sources on the Web.

3.2 Information Model

We have built a set of loosely coupled components that perform the individual processing steps. Each component's results are stored as RDF data in a central repository using the OpenRDF Sesame framework [14].

Figure 1 shows the information model. The central class is `pw:Ngram`, which represents the notion of an extracted string of N words. The model records every position of the ngram in the input text, and each occurrence of the same ngram in the same document is a single instance of the `pw:Ngram` class.

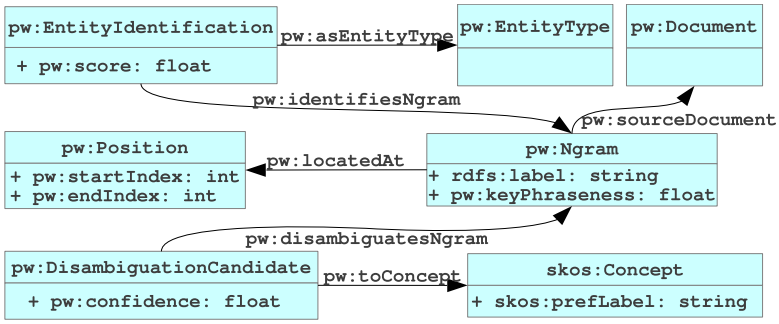

Fig. 1. Shared RDF model for ngram and entity information

The `pw:EntityType` class supports entity typing of ngrams. It has a fixed number of instances representing types such as people, organizations, locations, events, etc. In order to be able to record the relation between an ngram and its type, as well as an identification score reported by the extraction tool, the relation is modeled as an an object, of type `pw:EntityIdentification`.

[6] See a.o. `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`

`pw:DisambiguationCandidate` is introduced to allow ngrams to be annotated with corresponding concepts from external sources. This class records the relation (and the system's confidence in it) between an extracted ngram and an external source. These external sources are modeled as instances of `skos:Concept`. They are the building blocks of the taxonomy we generate.

Using a shared RDF model to hold extracted data ensures that components can interoperate and reuse each other's results. This is a significant advantage: it facilitates the use of different language processing tools in a single system by mapping their outputs to a common vocabulary. Moreover, users can add other Linked Data sources, and insert and remove processing steps, as they see fit. It can also be used for text annotation.[7]

In addition, the use of an RDF repository allows one to formulate SPARQL[8] queries over the aggregated data. Using these, data from different components can be analyzed quickly and efficiently at each processing step.

4 Generating the Taxonomy

Figure 2 shows the processing steps in our system, called F-STEP (Focused SKOS Taxonomy Extraction Process). Existing tools are used to extract entities and concepts from document text (steps 2a and 2b respectively in the Figure). Purpose-built components annotate entities with information contained in Linked Data sources (step 3), disambiguate concepts that are mapped to the same ngram (step 4), and consolidate concepts into a hierarchy (step 5).

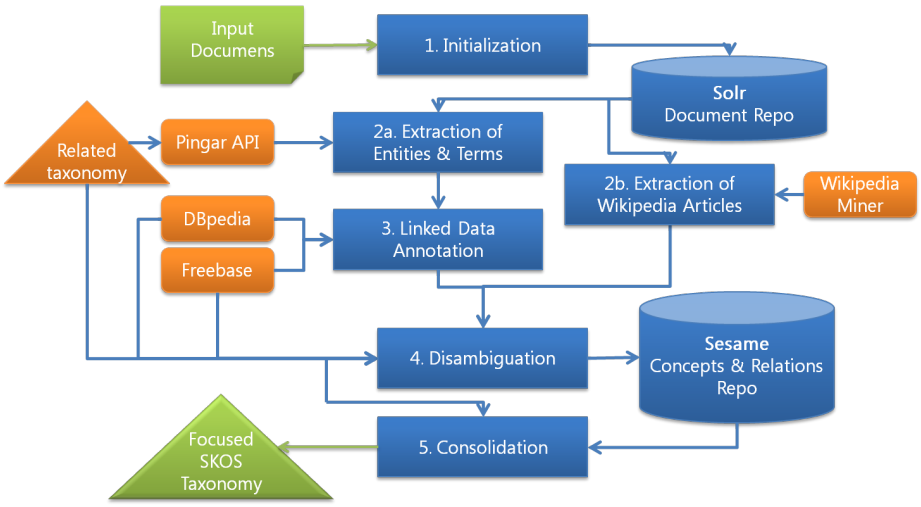

Fig. 2. Automated workflow for turning input documents into a focused taxonomy

[7] A possible alternative is the recently-defined NLP2RDF format `http://nlp2rdf.org`.
[8] See `http://www.w3.org/TR/sparql11-query/`

4.1 Initialization

Taxonomies organize knowledge that is scattered across documents. To federate inputs stored on file systems, servers, databases and document management systems, we use Apache Tika to extract text content from various file formats and Solr for scaleable indexing.[9] Solr stores multiple document collections in parallel, each document being referenced via a URL, which allows concepts to be linked back to the documents containing them in our RDF model.

4.2 Extracting Named Entities and Concepts

Extraction step 2a in Figure 2 uses a text analytics API[10] to identify names of people, organizations and locations, and to identify relevant terms in an existing taxonomy if one is supplied. Step 2b uses the Wikipedia Miner toolkit [15] to relate documents to relevant concepts in Wikipedia.

Named Entities. Names of people, organizations, and locations are concepts that can usefully be included in a taxonomy; existing systems extract such entities with an accuracy of 70%–80% [16]. We extract named entities from the input documents using the text analytics API and convert its response to RDF. Named entities are represented by a `pw:EntityIdentification` relation between the original ngram and an entity type. The entities are passed to the annotation step to disambiguate any matches to Linked Data concepts.

Concepts from Related Taxonomies. As mentioned in Section 3, the input can include one or more taxonomies from related domains. The same text analytics API records any concepts in a related taxonomy that appear in the input documents, maps them to SKOS, and links to the source document ngram via a `pw:DisambiguationCandidate` relation.

Concepts from Wikipedia. Each Wikipedia article is regarded as a "concept." Articles describe a single concept, and for (almost) any concept there exists a Wikipedia article. We use the Wikipedia Miner toolkit to annotate ngrams in the text with corresponding Wikipedia articles. This toolkit allows the number of annotations to be controlled, and disambiguates ngrams to their correct meaning—for example, the word *kiwi* may refer to a.o. a bird, a fruit, a person from NZ, or the NZ national rugby league team, all of which have distinct Wikipedia entries. The approach is described in detail in [15].

The user determines what kind of concepts will be included in the taxonomy. For example, if no related taxonomies are available, only named entities and Wikipedia articles returned by the Wikification process will be included in the final taxonomy.

[9] See `http://tika.apache.org/` and `http://lucene.apache.org/solr/`
[10] See `http://apidemo.pingar.com`

4.3 Annotating with Linked Data

Once entities such as people, places, and organisations have been extracted, the annotation step queries Freebase [17] and DBpedia [18] for corresponding concepts (Figure 2, step 3). The queries are based on the entity's type and label, which is the only structured information available at this stage. Other Linked Data sources can be consulted in this step, either by querying via a SPARQL endpoint,[11] which is how we consult DBpedia, or by accessing the Linked Data source directly over the HTTP protocol.

We define mappings of our three entity types to Linked Data concept classes. For example, in the case of Freebase, our entity type "Person" (`pw:person`) is mapped to `http://rdf.freebase.com/ns/people/person`, and for each extracted *person* entity Freebase is queried for lexically matching concepts of the mapped type. Several candidate concepts may be selected for each entity (the number is given as a configuration parameter). These matches are added as disambiguation candidates to every ngram that corresponds to the original entity.

4.4 Disambiguation

The preceding processing steps use various techniques to determine relevant concepts in documents. A direct consequence is that a given ngram may be mapped to more than one concept: a taxonomy term, a Wikipedia article, a Freebase or a DBpedia concept. Although the Wikipedia Miner incorporates its own built-in disambiguation component, this merely ensures that at most one Wikipedia concept corresponds to each ngram. A second disambiguation step (Figure 2, step 4) determines whether concepts from *different* sources share the same meaning and whether their meaning is contextually relevant.

The disambiguation is performed for each document, one ngram at a time. If an ngram has a single concept mapping, it is considered unambiguous and this concept is added to the final taxonomy. If an ngram has multiple mappings, the conflicting concepts are inspected first. Here, we compare the context of the ngram with the contexts of each concept, as it is defined in its original source. The context of the ngram is as a set of labels of concepts that co-occur in the same document, whereas the context of each concept is a set of labels derived from its associated concepts, computed in a way that depends on the concept's origin. In SKOS taxonomies, associated concepts are determined via `skos:broader`, `skos:narrower`, and `skos:related` relations. For each associated concept we collect the `skos:prefLabel` and one or more `skos:altLabels`. In Wikipedia, these labels are sourced from the article's redirects, its categories, the articles its abstract links to, and other linked articles whose semantic relatedness [15] exceeds a certain threshold (we used 0.3, which returns 27 linked articles on average). In the case of Freebase and DBpedia, we utilize the fact that many Freebase concepts have mappings to DBpedia, which in turn are (practically all) mapped to Wikipedia articles. We locate the corresponding Wikipedia article and use the above method to determine the concepts.

[11] A SPARQL endpoint is a web service that implements the W3C SPARQL protocol

Once all related labels have been collected we calculate the distance between every pair of labels. To account for lexical variation between the labels, we use the Dice coefficient between the sets of bigrams that represent the labels. We then compute a final similarity score by averaging the distance over the top n scoring pairs. n is chosen as the size of the smaller set, because if the concepts the sets represent are truly identical, every label in the smaller set should have at least one reasonably similar partner in the other set; larger values of n tend to dilute the similarity score when one of the concepts has many weakly associated concept labels, which is often the case for Wikipedia concepts.

Given this similarity metric, disambiguation proceeds as follows. First, we choose the concept with the greatest similarity to the ngram's context to be the canonical concept. (This assumes that there is at least one correct concept among the conflicting ones.) Second, we compare the similarity of every other candidate concept to the canonical one and, depending on its similarity score s, list it as an `skos:exactMatch` (if $s > 0.9$), an `skos:closeMatch` (if $0.9 \geq s \geq 0.7$), or discard it (if $s < 0.7$). The thresholds were determined empirically.

As an example of disambiguation, the ngram *oceans* matches three concepts: *Ocean*, *Oceanography* (both Wikipedia articles), and *Marine areas* (a taxonomy concept). The first is chosen as the canonical concept because its similarity with the target document is greatest. *Marine areas* is added as `skos:closeMatch`, because its similarity with *Ocean* is 0.87. However, *Oceanography*'s similarity falls below 0.7, so it is discarded. As a another example, the ngram *logged* is matched to both *Logs* (a taxonomy concept) and *Deforestation* (a Wikipedia article). *Logs* is semantically connected to another taxonomy concept, which is why it was not discarded by the text analytics API, but it is discarded by the disambiguation step because it is not sufficiently closely related to other concepts that occur in the same document.

4.5 Consolidation

The final step is to unite all unambiguous and disambiguated concepts found in documents into a single taxonomy. Each concept lists several URIs under `skos:exactMatch` and (possibly) `skos:closeMatch` that define it in other sources: the input taxonomy, Wikipedia, Freebase and DBpedia. These sources already organize concepts into hierarchies, but they differ in structure. The challenge is to consolidate these hierarchies into a single taxonomy.

Sources of Relations. Taxonomies from related domains, as optional inputs, already define the relations we seek: `skos:broader` and `skos:narrower`. However, they may cover certain areas in more or less detail than what we need, which implies that some levels should be flattened while others are expanded. Because *broader* and *narrower* are transitive relations, flattening is straightforward. For expansion, concepts from other sources are needed.

Wikipedia places its articles into categories. For example, the article on George Washington belongs to 30 categories; some useful, e.g. *Presidents of the*

US and *US Army generals*, and others that are unlikely to be relevant in a taxonomy, e.g. *1732 births*. Some articles have corresponding categories (e.g., there is a category "George Washington"), which lead to further broader categories. Furthermore, names may indicate multiple relations (e.g. *Politicians of English descent* indicates that *George Washington* is both a *Politician* and *of English descent*). Wikipedia categories tend to be fine-grained, and we discard information to create broader concepts. We remove years (*1980s TV series* becomes *TV series*), country and language identifiers (*American sitcoms* becomes *Sitcoms*; *Italian-language comedy films* becomes *Comedy films*), and verb and prepositional phrases that modify a head noun (*Educational institutions established in the 1850s* becomes *Educational institutions*; *Musicians by country* becomes *Musicians*). The entire Wikipedia category structure is available on DBpedia in SKOS format, which makes it easy to navigate. We query the SPARQL DBpedia endpoint to determine categories for a given Wikipedia article.

Other potential sources are Freebase, where categories are defined by users, and DBpedia, which extracts relations from Wikipedia infoboxes. We plan to use this information in future when consolidating taxonomies.

Consolidation Rules. F-STEP consolidates the taxonomy that has been generated so far using a series of rules. First, direct relations are added between concepts. For each concept with a SKOS taxonomy URI, if its broader and narrower concepts match other input concepts, we connect these concepts, e.g. *Air transport* `skos:narrower` *Fear of flying*. If a concept has a Wikipedia URI and its immediate Wikipedia categories match an existing concept, we connect these concepts, e.g. *Green tea* `skos:narrower` *Pu-erh tea*.

Following the intuition that some concepts do not appear in the documents, but may useful for grouping others that do, we iteratively add such concepts. For each concept with a SKOS taxonomy URI, we use a transitive SPARQL query to check whether it can be connected by new intermediate concepts to other concepts. If a new concept is found, it is added to the taxonomy and its relations are populated for all further concepts. For example, this rule connects concepts like *Music* and *Punk rock* via a new concept *Music genres*, whereupon a further relation is added between *Music genres* and *Punk rock*.

Next, the Wikipedia categories are examined to identify those of interest. The document collection itself is used to quantify the degree of interest: categories whose various children co-occur in many documents tend to be more relevant. Specifically, a category's "quality" is computed by iterating over its children and checking how many documents contain them. If this score, normalized by the total number of comparisons made, exceeds a given threshold, the category is added to the output taxonomy. This helps eliminate categories that combine too many concepts (e.g. *Living people* in a news article) or that do not group co-occurring concepts, and singles out useful categories instead (e.g. *Seven Summits* might connect *Mont Blanc*, *Puncak Jaya*, *Aconcagua*, and *Mount Everest*). Next, we retrieve broader categories for these newly added categories and check whether their names match existing concepts, allowing us to add new

relations. One could continue up the Wikipedia category tree, but the resulting categories are less satisfactory. For example, *Music* belongs to *Sound*, which in turn belongs to *Hearing*, but the relation between *Music* and *Hearing* is associative rather than hierarchical. In fact, unlike conventional SKOS taxonomies, the Wikipedia category structure is not, in general, transitive.

Parentheses following some Wikipedia article names indicate possible groupings for a concept, e.g. *Madonna_(entertainer)* is placed under *Entertainers*, if such a concept exists. We also match each category name's last word against existing concept names, but choose only the most frequent concepts to reduce errors introduced by this crude technique.

We group all named entities that are found in Freebase using the Freebase categories, and all those found in DBpedia using the corresponding Wikipedia categories. The remainder are grouped by their type, e.g. *John Doe* under *Person*.

These techniques tend to produce forests of small subtrees, because general concepts rarely appear in documents. We check whether useful general terms can be found in a related taxonomy, and also examine the small upper-level taxonomy that a user may provide, as mentioned in Section 1. For example, a media website may divide news into *Business*, *Technology*, *Sport* and *Entertainment*, with more specific areas underneath, e.g. *Celebrities*, *Film*, *Music*—a two-level taxonomy of broad categories. For each input concept we retrieve its broadest concept—the one below the root—and add it, skipping intermediate levels. This rule adds relations like *Cooperation* `skos:broader` *Business and industry*.

Pruning Heuristics. Prunning can make a taxonomy more usable, and eliminate redundancies. First, following [4], who extract a taxonomy from WordNet, we elide parent–child links for single children. If a concept has a single child that itself has one or more children, we remove the child and point its children directly to its parent.

Second, we eliminate multiple inheritance that repeats information in the same taxonomy subtree, which originates from redundancy in the Wikipedia category structure. We identify cases where either relations or concepts can be removed without compromising the tree's informativeness. Figure 3 shows examples. In (a) the two-parent concept *Manchester United FC* is reduced to a single parent by removing a node that does not otherwise contribute to the structure. In (b) the two-parent concept *Tax* is reduced to a single parent by removing a small redundant subtree. In (c) a common parent of the two-parent concepts *The Notorious B.I.G.* and *Tupac Shakur* is pruned.

5 Evaluation and Discussion

Domain-specific taxonomies (and ontologies) are typically evaluated by (a) comparing them to manually-built taxonomies, (b) evaluating the accuracy of their concepts and relations, and (c) soliciting feedback from experts in the field. This section evaluates our system's ability to generate a taxonomy from a news collection. We give an overview of the dataset used, compare the dimensions of the

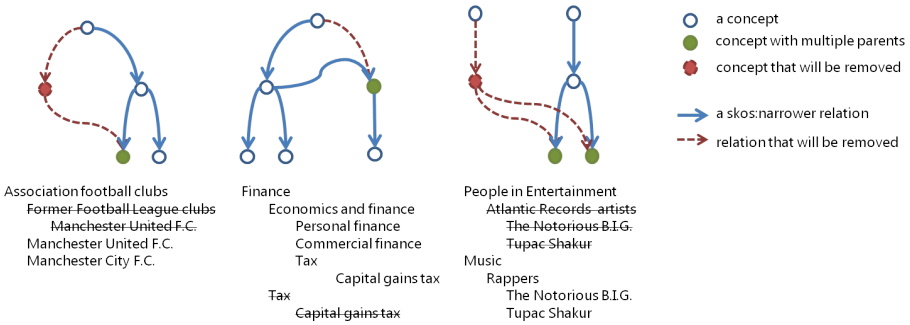

Fig. 3. Pruning concepts and relations to deal with multiple inheritance

taxonomy generated with other taxonomies, assess its coverage by comparing it with a hand-built taxonomy for the domain, and determine the accuracy of both its concepts and its relations with respect to human judgement.

5.1 The Domain

Fairfax Media is a large media organization that publishes hundreds of news articles daily. Currently, these are stored in a database, organized and retrieved according to manually assigned metadata. Manual assignment is time-consuming and error-prone, and automatically generated metadata, organized hierarchically for rapid access to news on a particular topic or in a general field, would be of great benefit.

We collected 2000 news articles (4.3MB of uncompressed text) from December 2011, averaging around 300 words each. We used the UK Integrated Public Service Sector vocabulary (`http://doc.esd.org.uk/IPSV/2.00.html`) as an input taxonomy. A taxonomy was extracted using the method described in Section 4 and can be viewed at `http://bit.ly/f-step`. It contains 10,150 concepts and 12,700 relations and is comparable in size to a manually-constructed taxonomy for news, the New York Times taxonomy (`data.nytimes.com`), which lists 10,400 *People*, *Organizations*, *Locations* and *Descriptors*. The average depth of the tree is 2.6, with some branches being 10 levels deep. Each concept appears in an average of 2 news articles. The most frequent, *New Zealand*, appears as metadata for 387 articles; the most topical, *Christmas*, is associated with 127 articles. About 400 concepts were added during the consolidation phase to group other concepts, and do not appear as metadata.

5.2 Coverage Comparison

To investigate the coverage of the automatically-generated taxonomy, we compared it with one comprising 458 concepts that Fairfax librarians had constructed manually to cover all existing and future news articles. Interestingly, this taxonomy was never completed, most likely because of the labor involved. Omissions

tend to be narrower concepts like individual sports, movie genres, music events, names of celebrities, and geographic locations. In order to evaluate our new taxonomy in terms of recall, we checked which of the 458 manually assigned concepts have labels that match labels in the new taxonomy (considering both preferred or alternative labels in both cases). There were a total of 271 such "true positives," yielding a recall of 59%. However, not all the manually assigned concepts are actually mentioned in the document set used to generate our taxonomy, and are therefore, by definition, irrelevant to it. We used Solr to seek concepts for which at least one preferred or alternative label appears in the document set, which reduced the original 458 concepts to 298 that are actually mentioned in the documents. Re-calculating the recall yields a figure of 75% (224 out of 298).

Inspection shows that some of the missing concepts are present but with different labels—instead of *Drunk*, the automatically generated taxonomy includes *Drinking alcohol* and *Alcohol use and abuse*. Others are present in a more specific form—instead of *Ethics* it lists *Ethical advertising* and *Development ethics*. Nevertheless, some important concepts are missing—for example, *Immigration*, *Laptop* and *Hospitality*.

5.3 Accuracy of Concepts

Fifteen human judges were used to evaluate the precision of the concepts present in the taxonomy generated from the documents. Each judge was presented with the text of a document and the taxonomy concepts associated with it, and asked to provide yes/no decisions on whether the document refers to each term. Five documents were chosen and given to all judges; a further 300 documents were distributed equally between the judges.

Looking first at the five common documents, the system extracted 5 to 30 concepts from each, with an average of 16. Three judges gave low scores, agreeing with only 74%, 86% and 90% of the concepts respectively, averaged over the five documents. The remaining 12 each agreed with virtually all—more than 97%—of the concepts identified by the system. The overall precision for automatic identification of concepts, averaged over all 15 judges, was 95.2%.

Before these figures were calculated the data was massaged slightly to remove an anomaly. It turned out that the system identified for each article the name of the newspaper in which it was published (e.g. *Taranaki Daily News*), but the human judges disagreed with one another on whether that should be counted as a valid concept for the article. A decision was taken to exclude the name of the newspaper from the first line of the article.

Turning now to the 300 documents that were examined by one judge each, the system identified a total of 3,347 concepts. Of these, 383 were judged incorrect, yielding an overall precision of 88.6%. (In 15 cases the judge was unwilling to give a yes/no answer; these were counted as incorrect.) Table 1 shows the source of the errors. Note that any given concept may originate in more than one source, which explains the discrepancy in the total of the Errors column (393, not 383). The most accurate concepts are ones that describe people. The most error-prone ones emanate from the input taxonomy, 26% of which are incorrect. This taxonomy

Table 1. Sources of error in concept identification

Type	Number	Errors	Rate
People	1145	37	3.2%
Organizations	496	51	10.3%
Locations	988	114	11.5%
Wikipedia named entities	832	71	8.5%
Wikipedia other entities	99	16	16.4%
Taxonomy	868	229	26.4%
DBPedia	868	81	8.1%
Freebase	135	12	8.9%
Overall	3447	393	11.4%

describes rather general concepts, which introduces more ambiguity than the other sources.

5.4 Accuracy of Relations

The same fifteen judges were used to evaluate the precision of the hierarchical relations present in the taxonomy. Each judge received 100 concept pairs and was asked for a yes/no decision as to whether that relation makes sense—i.e., whether the first concept really is narrower than the second. A total of 750 relations were examined, each adjudicated by two different judges.

The overall precision figure was 90%—that is, of the 1500 decisions, judges expressed disagreement in 150 cases. The interannotator agreement, calculated as the number of relationships that both judges agreed on expressed as a proportion of all relationships, was 87%.

An examination of where the two judges made different decisions revealed that some were too strict, or simply wrong (for example, *Acid* $\sqsubset$ *base chemistry*, *Leeds* $\sqsubset$ *North Yorkshire*, *History of Israel* $\sqsubset$ *Israel*, where $\sqsubset$ means "has parent"). Indeed, it appears that, according to some judges, polio is not an infectious disease and Sweden is not in Scandinavia! It is interesting to analyze the clear errors, discarding cases where the judges conflicted. Of the 25 situations where both judges agreed that the system was incorrect, ten pairs were related but not in a strict hierarchical sense (e.g., *Babies* $\not\sqsubset$ *school children*), four were due to an overly simplistic technique that we use to identify the head of a phrase (e.g. *Daily Mail* $\not\sqsubset$ *Mail*), two could have (and should have) been avoided (e.g. *League* $\not\sqsubset$ *League*), and nine were clearly incorrect and correspond to bugs that deserve further investigation (e.g. *Carter Observatory* $\not\sqsubset$ *City*).

6 Conclusions

This paper has presented a new approach to analyzing documents and generating taxonomies focused on their content. It combines existing tools with new

techniques for disambiguating concepts originating from various sources and consolidating them into a single hierarchy. A highlight of the scheme is that it can be easily extended. The use of RDF technology and modeling makes coupling and reconfiguring the individual components easy and flexible. The result, an SKOS taxonomy that is linked to both the documents and Linked Data sources, is a powerful knowledge organization structure that can serve many tasks: browsing documents, fueling facetted search refinements, question answering, finding similar documents, or simply analyzing one's document collection.

The evaluation has shown that in one particular scenario in the domain of news, the taxonomy that is generated is comparable to manually built exemplars in the dimensions of the hierarchical structure and in its coverage of the relevant concepts. Recall of 75% was achieved with respect to a manually generated taxonomy for the same domain, and inspection showed that some of the apparently missing concepts are present but with different—and arguably more precise—labels. With respect to multiple human judgements on five documents, the accuracy of concepts exceeded 95%; the figure decreased to 89% on a larger dataset of 300 documents. The accuracy of relations was measured at 90% with respect to human judgement, but this is diluted by human error. Analysis of cases where two judges agreed that the system was incorrect revealed that at least half were anomalies that could easily be rectified in a future version. Finally, although we still plan to perform an evaluation in an application context, initial feedback from professionals in the news domain is promising. Some professionals expect to tweak the taxonomy manually by renaming some top concepts, removing some irrelevant relations, or even re-grouping parts of the hierarchy, and we have designed a user interface that supports this.

Compared to the effort required to come up with a taxonomy manually, a cardinal advantage of the automated system is speed. Given 10,000 news articles, corresponding to one week's output of Fairfax Media, a fully-fledged taxonomy is generated in hours. Another advantage is that the taxonomy focuses on what actually appears in the documents. Only relevant concepts and relations are included, and relations are created based on salience in the documents (e.g. occurrence counts) rather than background knowledge. Finally, because Wikipedia and Freebase are updated daily by human editors, the taxonomy that is produced is current, which is important for ever-changing domains such as news.

Finally, the approach is applicable to any domain. Every knowledge-based organization deals with mountains of documents. Taxonomies are considered a very useful document management tool, but uptake has been been slow due to the effort involved in building and maintaining them. The scheme described in this paper reduces that cost significantly.

Acknowledgements. This work was co-funded by New Zealand's Ministry of Science and Innovation. We also thank David Milne and Shane Stuart from the University of Waikato and Reuben Schwarz from Fairfax Media NZ.

References

1. Caraballo, S.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proc. of the 37th Annual Meeting of the ACL, ACL (1999) 120–126
2. Ponzetto, S., Strube, M.: Deriving a large scale taxonomy from wikipedia. In: Proc. of the 22nd National Conference on Artificial Intelligence, AAAI Press (2007) 1440–1445
3. Snow, R., Jurafsky, D., Ng, A.: Semantic taxonomy induction from heterogenous evidence. In: Proc. of the 21st Intl. Conf. on Computational Linguistics, ACL (2006) 801–808
4. Stoica, E., Hearst, M.A.: Automating creation of hierarchical faceted metadata structures. In: In Procs. of the HLT/NAACL Conference. (2007)
5. Dakka, W., Ipeirotis, P.: Automatic extraction of useful facet hierarchies from text databases. In: Proc. of the 24th IEEE Intl. Conf. on Data Engineering, IEEE (2008) 466–475
6. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proc. of the 22nd Annual Intl. Conf. on R&D in Information Retrieval, ACM (1999) 206–213
7. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proc. of the 14th Conference on Computational linguistics, ACL (1992) 539–545
8. Suchanek, F., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proc. of the 16th Intl. Conference on World Wide Web, ACM (2007) 697–706
9. Wu, W., Li, H., Wang, H., Zhu, K.: Probase: A probabilistic taxonomy for text understanding. In: Proc. of the 2012 ACM Intl. Conf. on Management of Data, ACM (2012) 481–492
10. Matuszek, C., Witbrock, M., Kahlert, R., Cabral, J., Schneider, D., Shah, P., Lenat, D.: Searching for common sense: Populating cyc from the web. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence, AAAI Press (2005) 1430–1435
11. Milne, D., Witten, I.: Learning to link with wikipedia. In: Proc. of the 17th Conference on Information and Knowledge Management, ACM (2008) 509–518
12. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proc. of the 7th Intl. Conf. on Semantic Systems, ACM (2011) 1–8
13. Augenstein, I., Pado, S., Rudolph, S.: LODifier: Generating Linked Data from Unstructured Text. In: Proc. of the 9th Extended Semantic Web Conference (ESWC 2012). Number 7295 in LNCS, Springer Verlag, Heidelberg (2012) 210–224
14. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Proc. of the 1st Intl. Semantic Web Conference. Number 2342 in LNCS, Springer Verlag, Heidelberg Germany (2002) 54–68
15. Milne, D., Witten, I.H.: An open-source toolkit for mining Wikipedia. Artificial Intelligence (2012)
16. Marrero, M., Sanchez-Cuadrado, S., Lara, J., Andreadakis, G.: Evaluation of Named Entity Extraction Systems. In: Proc. of the Conference on Intelligent Text Processing and Computational Linguistics, CICLing. (2009) 47–58
17. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proc. of Intl. Conf. on Management of Data. SIGMOD '08, New York, NY, USA, ACM (2008) 1247–1250
18. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: In 6th Intl. Semantic Web Conference, Busan, Korea, Springer (2007) 11–15