Automating Creation of Hierarchical Faceted Metadata Structures

Emilia Stoica and **Marti A. Hearst**
School of Information
UC Berkeley, Berkeley, CA
estoica,hearst@ischool.berkeley.edu

Megan Richardson
Department of Mathematical Sciences
NMSU, Las Cruces, NM
merichar@nmsu.edu

Abstract

We describe Castanet, an algorithm for automatically generating hierarchical faceted metadata from textual descriptions of items, to be incorporated into browsing and navigation interfaces for large information collections. From an existing lexical database (such as WordNet), Castanet carves out a structure that reflects the contents of the target information collection; moderate manual modifications improve the outcome. The algorithm is simple yet effective: a study conducted with 34 information architects finds that Castanet achieves higher quality results than other automated category creation algorithms, and 85% of the study participants said they would like to use the system for their work.

1 Introduction

It is becoming widely accepted that the standard search interface, consisting of a query box and a list of retrieved items, is inadequate for navigation and exploration in large information collections such as online catalogs, digital libraries, and museum image collections. Instead, user interfaces which organize and group retrieval results have been shown to be helpful for and preferred by users over the straight results-list model when engaged in exploratory tasks (Yee et al., 2003; Pratt et al., 1999; Kaki, 2005). In particular, a representation known as hierarchical faceted metadata is gaining great traction within the information architecture and enterprise search communities (Yee et al., 2003; Weinberger, 2005).

A considerable impediment to the wider adoption of collection navigation via metadata in general, and hierarchical faceted metadata in particular, is the need to create the metadata hierarchies and assign the appropriate category labels to the information items. Usually, metadata category structures are manually created by information architects (Rosenfeld and Morville, 2002). While manually created metadata is considered of high quality, it is costly in terms of time and effort to produce, which makes it difficult to scale and keep up with the vast amounts of new content being produced.

In this paper, we describe Castanet, an algorithm that makes considerable progress in automating faceted metadata creation. Castanet creates domain-specific overlays on top of a large general-purpose lexical database, producing surprisingly good results in a matter of minutes for a wide range of subject matter.

In the next section we elaborate on the notion of hierarchical faceted metadata and show how it can be used in interfaces for navigation of information collections. Section 3 describes other algorithms for inducing category structure from textual descriptions. Section 4 describes the Castanet algorithm, Section 5 describes the results of an evaluation with information architects, and Section 6 draws conclusions and discusses future work.

2 Hierarchical Faceted Metadata

A hierarchical faceted metadata system (HFC) creates a set of category hierarchies, each of which corresponds to a different facet (dimension or type). The main application of hierarchical faceted metadata is in user interfaces for browsing and navigating collections of like items.

In the case of a recipe collection, for example, facets may consist of dish type (salad, appetizer), ingredients such as fruits (apricot, apple), vegetables (broccoli, cabbage), meat (beef, fish), preparation method (fry, bake, etc.), calorie count, and so on. Decomposing the description into independent categories allows users to move through large information spaces in a flexible manner. The category metadata guides the user toward possible choices, and organizes the results of keyword searches, allowing users to both refine and expand the current query, while maintaining a consistent representation of the collection's structure. This use of metadata should be integrated with free-text search, allowing the user to follow links, then add search terms, then follow more links, without interrupting the interaction flow.

Usability studies have shown that, when incorporated into a properly-designed user interface, hierarchical faceted metadata provides a flexible, intuitive way to explore a large collection of items that enhances feelings of discovery without inducing a feeling of being lost (Yee et al., 2003).

Note that the HFC representation is intermediate in complexity between that of a monolithic hierarchy and a full-blown ontology. HFC does not capture relations and inferences that are essential for some applications. For example, faceted metadata can express that an image contains a hat and a man and a tree, and perhaps a wearing activity, but does not indicate who is wearing what. This relative simplicity of representation suggests that automatically inferring facet hierarchies may be easier than the full ontology inference problem.

3 Related Work

There is a large literature on document classification and automated text categorization (Sebastiani, 2002). However, that work assumes that the categories of interest are already known, and tries to assign documents to categories. In contrast, in this paper we focus on the problem of determining the categories of interest.

Another thread of work is on finding synonymous terms and word associations, as well as automatic acquisition of IS-A (or genus-head) relations from dictionary definitions and free text (Hearst, 1992; Caraballo, 1999). That work focuses on finding the right position for a word within a lexicon, rather than building up comprehensible and coherent faceted hierarchies.

A major class of solutions for creating subject hierarchies uses data clustering. The Scatter/Gather system (Cutting et al., 1992) uses a greedy global agglomerative clustering algorithm where an initial set of k clusters is recursively re-clustered until only documents remain. Hofmann (1999) proposes the probabilistic latent semantic analysis algorithm (pLSA), a probabilistic version of clustering that uses latent semantic analysis for grouping words and annealed EM for model fitting.

The greatest advantage of clustering is that it is fully automatable and can be easily applied to any text collection. Clustering can also reveal interesting and potentially unexpected or new trends in a group of documents. The disadvantages of clustering include their lack of predictability, their conflation of many dimensions simultaneously, the difficulty of labeling the groups, and the counter-intuitiveness of cluster sub-hierarchies (Pratt et al., 1999).

Blei et al. (2003) developed the LDA (Latent Dirichlet Allocation) method, a generative probabilistic model of discrete data, which creates a hierarchical probabilistic model of documents. It attempts to analyze a text corpus and extract the topics that combined to form its documents. The output of the algorithm was evaluated in terms of perplexity reduction but not in terms of understandability of the topics produced.

Sanderson and Croft (1999) propose a method called *subsumption* for building a hierarchy for a set of documents retrieved for a query. For two terms x and y, x is said to subsume y if the following conditions hold: $P(x|y) \geq 0.8, P(y|x) < 1$. In other words, x subsumes y and is a parent of y, if the documents which contain y, are a subset of the documents which contain x. To evaluate the algorithm the authors asked 8 participants to look at parent-child pairs and state whether or not they were "interesting". Participants found 67% to be interesting as compared to 51% for randomly chosen pairs of words. Of those interesting pairs, 72% were found to display a "type-of" relationship.

Nevill-Manning et.al (1999), Anick et.al (1999) and Vossen (2001) build hierarchies based on substring inclusion. For example, the category *full text indexing and retrieval* is the child of *indexing and retrieval* which in turn is the child of *index*. While these string inclusion approaches expose some structure of the dataset, they can only create subcategories which are substrings of the parent category, which is very restrictive.

Another class of solutions make use of existing lexical hierarchies to build category hierarchies, as we do in this paper. For example, Navigli and Velardi (2003) use WordNet (Fellbaum, 1998) to build a complex ontology consisting of a wide range of relation types (demonstrated on a travel agent domain), as opposed to a set of human-readable hierarchical facets. They develop a complex algorithm for choosing among WordNet senses; it requires building a rich semantic network using WordNet glosses, meronyms, holonyms, and other lexical relations, and using the semantically annotated SemCor collection. The semantic nets are intersected and the correct sense is chosen based on a score assigned to each intersection. Mihalcea and Moldovan (2001) describe a sophisticated method for simplifying WordNet in general, rather than tailoring it to a specific collection.

4 Method

The main idea behind the Castanet algorithm[1] is to carve out a structure from the hypernym (IS-A) relations within the WordNet (Fellbaum, 1998) lexical database. The primary unit of representation in WordNet is the synset, which is a set of words that are considered synonyms for a particular concept. Each synset is linked to other synsets via several types of lexical and semantic relations; we only use hypernymy (IS-A relations) in this algorithm.

[1] A simpler, un-evaluated version of this algorithm was presented previously in a short paper (Stoica and Hearst, 2004).

4.1 Algorithm Overview

The Castanet algorithm assumes that there is text associated with each item in the collection, or at least with a representative subset of the items. The textual descriptions are used *both* to build the facet hierarchies and to assign items (documents, images, citations, etc.) to the facets. The text does not need to be particularly coherent for the algorithm to work; we have applied it to fragmented image annotations and short journal titles, but if the text is impoverished, the information items will not be labeled as thoroughly as desirable and additional manual annotation may be needed.

The algorithm has five major steps:

1. Select target terms from textual descriptions of information items.
2. Build the Core Tree:
   - For each term, if the term is unambiguous (see below), add its synset's IS-A path to the Core Tree.
   - Increment the counts for each node in the synset's path with the number of documents in which the target term appears.
3. Augment the Core Tree with the remaining terms' paths:
   - For each candidate IS-A path for the ambiguous term, choose the path for which there is the most document representation in the Core Tree.
4. Compress the augmented tree.
5. Remove top-level categories, yielding a set of facet hierarchies.

We describe each step in more detail below.

4.2 Select Target Terms

Castanet selects only a subset of terms, called *target terms*, that are intended to best reflect the topics in the documents. Similarly to Sanderson and Croft (1999), we use the *term distribution* – defined as the number of item descriptions containing the term – as the selection criterion. The algorithm retains those terms that have a distribution larger than a threshold and eliminates terms on a stop list. One and two-word consecutive noun phrases are eligible to be considered as terms. Terms that can be adjectives or verbs as well as nouns are optionally deleted.

4.3 Build the Core Tree

The Core Tree acts as the "backbone" for the final category structure. It is built by using paths derived from unambiguous terms, with the goal of biasing the final structure towards the appropriate senses of words.

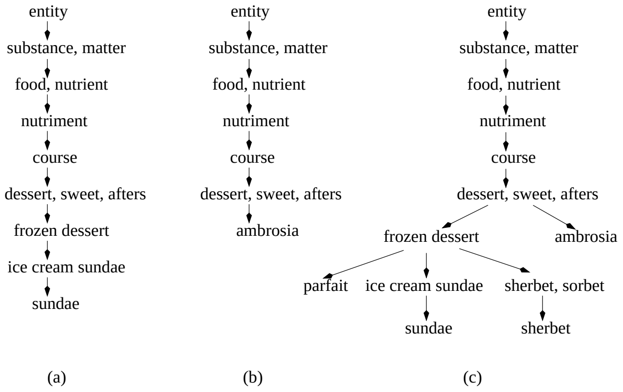


Figure 1: Merging hypernym paths.

4.3.1 Disambiguate using Wordnet Domains

A term is considered unambiguous if it meets at least one of two conditions:

(1) The term has only one sense within WordNet, or
(2) (Optional) The term matches one of the pre-selected WordNet domains (see below).

From our experiments, about half of the eligible terms have only one sense within WordNet. For the rest of terms, we disambiguate between multiple senses as follows.

WordNet provides a cross-categorization mechanism known as *domains*, whereby some synsets are assigned general category labels. However, only a small subset of the nouns in WordNet have domains assigned to them. For example, for a medicine collection, we found that only 4% of the terms have domains *medicine* or *biology* associated with them. For this reason, we use an additional resource called *Wordnet Domains* (Magnini, 2000), which assigns domains to WordNet synsets. In this resource, *every* noun synset in WordNet has been semi-automatically annotated with one of about 200 Dewey Decimal Classification labels. Examples include *history, literature, plastic arts, zoology,* etc.

In Castanet, Wordnet Domains are used as follows. First, the system counts how many times each domain is represented by target terms, building a list of the most well-represented domains for the collection. Then, in a manual intervention step, the information architect selects the subset of the well-represented domains which are meaningful for the collection in question.

For example, for a collection of biomedical journal titles, *Surgery* should be selected as a domain, whereas for an art history image collection, *Architecture* might be chosen. When processing the word *lancet*, the choice of domain distinguishes between the hyponym path *entity* → *object* → *artifact* → *instrumentality* → *device* → *instrument* → *medical instrument* → *surgical instrument*

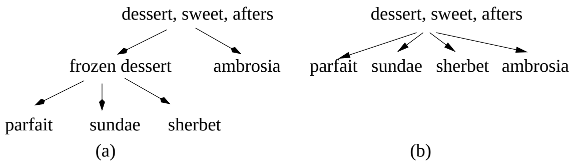


Figure 2: Compressing the tree.

→ *lancet* and *entity* → *object* → *artifact* → *structure, construction* → *arch* → *pointed arch* → *Gothic arch* → *lancet arch, lancet* → *lancet*.

In some cases, more than one domain may be relevant for a given term and for a given collection. For example, the term *brain* is annotated with two domains, *Anatomy* and *Psychology*, which are both relevant domains for a biomedical journal collection. Currently for these cases the algorithm breaks the tie by choosing the sense with the lowest WordNet sense number (corresponding to the most common sense), which in this case selects the *Anatomy* sense. However, we see this forced choice as a limitation, and in future work we plan to explore how to allow a term to have more than one occurrence in the metadata hierarchies.

4.3.2 Add Paths to Core Tree

To build the Core Tree, the algorithm marches down the list of unambiguous terms and for each term looks up its synset and its hypernym path in WordNet. (If a term does not have representation in WordNet, then it is not included in the category structure.) To add a path to the Core Tree, its path is merged with those paths that have already been placed in the tree. Figure 1(a-b) shows the hypernym paths for the synsets corresponding to the terms *sundae* and *ambrosia*. Note that they have several hypernym path nodes in common: *(entity), (substance, matter), (food, nutrient), (nutriment), (course), (dessert, sweet, afters)*. Those shared paths are merged by the algorithm; the results, along with the paths for *parfait* and *sherbert* are shown in Figure 1(c).

In addition to augmenting the nodes in the tree, adding in a new term increases a count associated with each node on its path; this count corresponds to how many documents the term occurs in. Thus the more common a term, the more weight it places on the path it falls within.

4.4 Augment the Core Tree / Disambiguate Terms

The Core Tree contains only a subset of terms in the collection (those that have only one path or whose sense can be selected with WordNet Domains). The next step is to add in the paths for the remaining target terms which are ambiguous according to WordNet.

The Core Tree is built with a bias towards paths that are most likely to be appropriate for the collection as a whole. When confronted with a term that has multiple possible IS-A paths corresponding to multiple senses, the system favors the more common path over other alternatives.

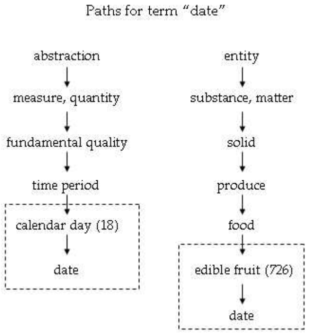


Figure 3: Two path choices for an ambiguous term.

Assume that we want to add the term *date* to the Core Tree for a collection of recipes, and that currently there are two paths corresponding to two of its senses in the Core Tree (see Figure 3). To decide which of the two paths to merge *date* into, the algorithm looks at the number of items assigned to the deepest node that is held in common between the existing Core Tree and each candidate path for the ambiguous term. The path for the *calendar day* sense has fewer than 20 documents assigned to it (corresponding to terms like *Valentine's Day*), whereas the path for the *edible fruit* sense has more than 700 documents assigned. Thus *date* is added to the fruit sense path. (The counts for the ambiguous terms' document hits are *not* incorporated into the new tree.)

Also, to eliminate unlikely senses, each candidate sense's hypernym path is required to share at least $j\%$ of its nodes with nodes already in the Core Tree, where the user sets j (usually between 40 and 60%). Thus the romantic appointment sense of *date* would not be considered as most of its hypernym path is not in the Core Tree. If no path passes the threshold, then the first sense's hypernym path (according to WordNet's sense ordering) is placed in the tree.

4.5 Compress the Tree

The tree that is obtained in the previous step usually is very deep, which is undesirable from a user interface perspective. Castanet uses two rules for compressing the tree:

1. Starting from the leaves, recursively eliminate a parent that has fewer than *k* children, unless the parent is the root or has an item count larger than 0.1×(maximum term distribution).
2. Eliminate a child whose name appears within the parent's name, unless the child contains a WordNet domain name.

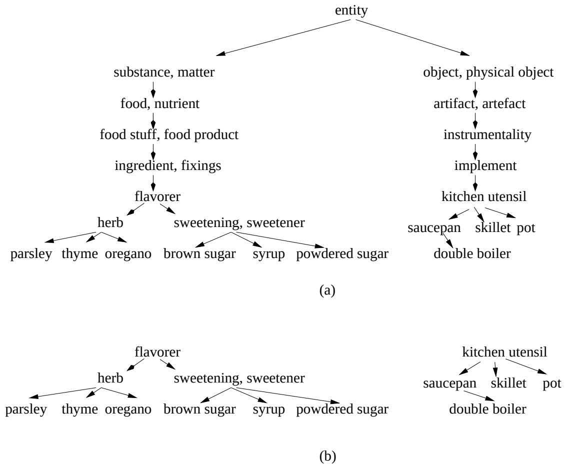


Figure 4: Eliminating top levels.

For example, consider the tree in Figure 1(c) and assume that $k = 2$, which means eliminate parents that have fewer than two children.

Starting from the leaves, by applying Rule 2, nodes (*ice cream sundae*), (*sherbet, sorbet*), (*course*), (*nutriment*), (*food, nutrient*), (*substance, matter*) and (*entity*) are eliminated since they have only one child. Figure 2(a) shows the resulting tree. Next, by applying Rule 3, the node *frozen dessert* is eliminated, since it contains the word *dessert* which also appears in the name of its parent. The final tree is presented in Figure 2(b). Note that this is a rather aggressive compression strategy, and the algorithm can be adjusted to allow more hierarchy to be retained.

4.6 Prune Top Level Categories / Create Facets

The final step is to create a set of facet sub-hierarchies. The goal is to create a moderate set of facets, each of which has moderate depth and breadth at each level, in order to enhance the navigability of the categories. Pruning the top levels can be automated, but a manual editing pass over the outcome will produce the best results.

To eliminate the top levels in an automated fashion, for each of the nine tree roots in the WordNet noun database, manually cut the top t levels (where $t = 4$ for the recipes collection). Then, for each of the resulting trees, recursively test if its root has more than $n = 6$ children. If it does, then the tree is considered a facet; otherwise, the current root is deleted and the algorithm tests to see if each new root has n children. Those subtrees that do not meet the criterion are omitted from the final set of facets.

Consider the tree in Figure 4(a). In this case, the categories of interest are (*flavorer*) and (*kitchen utensil*) along with their children. However, to reach any of these categories, the user has to descend six levels, each of which has very little information. Figure 4(b) shows the resulting facets, which (subjectively) are at an informative level of description for an information architecture. (In this illustration, $t = 2$.)

Often the internal nodes of WordNet paths do not have the most felicitous names, e.g., *edible fruit* instead of *fruit*. Although we did not edit these names for the usability study, it is advisable to do so.

5 Evaluation

The intended users of the Castanet algorithm are information architects and others who need to build structures for information collections. A successful algorithm must be perceived by information architects as making their job easier. If the proposed category system appears to require a lot of work to modify, then IAs are likely to reject it. Thus, to evaluate Castanet's output, we recruited information architects and asked them to compare it to one other state-of-the-art approach as well as a baseline. The participants were asked to assess the qualities of each category system and to express how likely they would be to use each in their work.

5.1 Study Design

The study compared the output of four algorithms: (a) Baseline (frequent words and two-word phrases), (b) Castanet, (c) LDA (Blei et al., 2003)[2] and (d) Subsumption (Sanderson and Croft, 1999). The algorithms were applied to a dataset of $13,000$ recipes from Southwestcooking.com. Participants were recruited via email and were required to have experience building information architectures and to be at least familiar with recipe websites (to show their interest in the domain).

Currently there are no standard tools used by information architects for building category systems from free text. Based on our own experience, we assumed a strong baseline would be a list of the most frequent words and two-word phrases (stopwords removed); the study results confirmed this assumption. The challenge for an automated system is to be preferred to the baseline.

The study design was within-participants, where each participant evaluated Castanet, a Baseline approach, and either Subsumption (N=16) or LDA (N=18).[3] Order of showing Castanet and the alternative algorithm was counterbalanced across participants in each condition.

Because the algorithms produce a large number of hierarchical categories, the output was shown to the

[2] Using code by Blei from www.cs.princeton.edu/˜blei/lda-c/

[3] Pilot studies found that participants became very frustrated when asked to compare LDA against Subsumption, since neither tested well, so we dropped this condition. We did not consider asking any participant to evaluate all three systems, to avoid fatigue. To avoid biasing participants towards any approach, the target algorithms were given the neutral names of Pine, Birch, and Oak. Castanet was run without Domains for a fairer comparison. Top level pruning was done automatically as described, but with a few manual adjustments.

	Cas.	Bas.	LDA	Cas.	Bas.	Sub.
Def. Yes	4	2	0	2	2	0
Yes	10	10	0	13	11	6
No	2	2	2	1	3	2
Def. No	2	4	16	0	0	8

Table 1: Responses to the question "Would you be likely to use this algorithm in your work?" comparing Castanet to the Baseline and LDA (N=18), and comparing Castanet to the Baseline and Subsumption (N=16).

	Cas. (34)	LDA (18)	Sub. (16)
Meaningful	2.9	1.2	1.8
Systematic	2.8	1.4	1.8
Import. Concepts	2.8	1.3	1.9

Table 2: Average responses to questions about the quality of the category systems. N shown in parentheses. Assessed on a four point scale where higher is better.

participants using the open source Flamenco collection browser[4] (see Figure 5). Clicking on a link shows subcategories as well as items that have been assigned that category. For example, clicking on the *Penne* subcategory beneath *Pasta* in the Castanet condition shows 5 recipes that contain the word *penne* as well as the other categories that have been assigned to these recipes. Since LDA does not create names for its output groups, they were assigned the generic names Category 1, 2, etc. Assignment of categories to items was done on a strict word-match basis; participants were not asked to assess the item assignment aspect of the interface.

At the start of the study, participants answered questions about their experience designing information architectures. They were then asked to look at a partial list of recipes and think briefly about what their goals would be in building a website for navigating the collection.

Next they viewed an ordered list of frequent terms drawn automatically from the collection (Baseline condition). After this, they viewed the output of one of the two target category systems. For each algorithm, participants were asked questions about the top-level categories, such as *Would you add any categories?* (possible responses: (a) No, None, (b) Yes, one or two, (c) Yes, a few, and (d) Yes, many). They were then asked to examine two specific top level categories in depth (e.g., *For category Bread, would you remove any subcategories?*). At the end of each assessment, they were asked to comment on general aspects of the category system as a whole (discussed below). After having seen both category systems, participants were asked to state how likely they would be to use the algorithm (e.g., *Would you use Oak? Would you use Birch? Would you use the frequent words list?*) Answer types were (a) No, definitely not, (b) Probably not, (c) Yes, I might want to use this system in some cases, and (d) Yes, I would definitely use this system.

5.2 Results

Table 1 shows the responses to the final question about how likely the participants are to use the results of each algorithm for their work. Both Castanet and the Baseline fare well, with Castanet doing somewhat better. 85% of the Castanet evaluators said yes or definitely yes to using it, compared to 74% for the Baseline. Only one participant said "no" to Castanet but "yes" to the Baseline, suggesting that both kinds of information are useful for information architects.

The comparison algorithms did poorly. Subsumption received 38% answering "yes" or "definitely yes" to the question about likelihood of use. LDA was rejected by all participants. A t-test (after converting responses to a 1-4 scale) shows that Castanet obtains significantly better scores than LDA ($t = 7.88 > 2.75$) and Subsumption ($t = 4.50 > 2.75$), for $p = 0.005$. The differences between Castanet and the Baseline are not significant.

Table 2 shows the average responses to the questions *(i) Overall, these are categories meaningful; (ii) Overall, these categories describe the collection in a systematic way; (iii) These categories capture the important concepts.*) They were scored as 1= Strongly disagree, 2 = Disagree Somewhat, 3 = Agree Somewhat, and 4 = Strongly agree. Castanet's score was about 35% higher than Subsumption's, and about 50% higher than LDA's.

Participants were asked to scrutinize the top-level categories and assess whether they would add categories, remove some, merge or rename some. The ratings were again converted to a four point scale (no changes = 4, change one or two = 3, change a few = 2, change many = 1). Table 3 shows the results. Castanet scores as well as or better than the others on all measures except Rename; Subsumption scores slightly higher on this measure, and does well on Split as well, but very poorly on Remove, reflecting the fact that it produces well-named categories at the top level, but too many at too fine a granularity.

Participants were also asked to examine two subcategories in detail. Table 4 shows results averaged across the two subcategories for number of categories to add, remove, promote, move, and how well the subcategories matched their expectations. Castanet performs especially well on this last measure (2.5 versus 1.5 and 1.7). Participants generally did not suggest moves or promotions.

Thus on all measures, we see Castanet outperforming the other state-of-the-art algorithms. Note that we did not explicitly evaluate the "facetedness" of the category systems, as we thought this would be too difficult for the participants to do. We feel the questions about the coher-

[4] Available at flamenco.berkeley.edu

	Cas. (34).	LDA (18)	Sub. (16)
Add	2.8	2.6	2.0
Remove	2.3	2.4	1.9
Rename	2.7	2.7	3.3
Merge	2.7	2.5	2.4
Split	3.8	3.3	3.8

Table 3: Assessing top-level categories.

	Cas. (34).	LDA (18)	Sub. (16)
Add	2.8	2.8	2.4
Remove	3.4	2.2	2.5
Promote	3.7	3.4	3.8
Move	3.8	3.3	3.6
Matched Exp.	2.5	1.5	1.7

Table 4: Assessing second-level categories.

ence, systematicity, and coverage of the category systems captured this to some degree.

6 Conclusions and Future Work

We have presented an algorithm called Castanet that creates hierarchical faceted metadata using WordNet and Wordnet Domains. A questionnaire revealed that 85% information architects thought it was likely to be useful, compared to 0% for LDA and 38% for Subsumption. Although not discussed here, we have successfully applied the algorithm to other domains including biomedical journal titles and art history image descriptions, and to another lexical hierarchy, MeSH.[5]

Although quite useful "out of the box," the algorithm could benefit by several improvements and additions. The processing of the terms should recognize spelling variations (such as aging vs. ageing) and morphological variations. Verbs and adjectives are often quite important for a collection (e.g., stir-fry for cooking) and should be included, but with caution. Some terms should be allowed to occur with more than one sense if this is required by the dataset (and some in more than one facet even with the same sense, as seen in the *brain* example). Currently if a term is in a document it is assumed to use the sense assigned in the facet hierarchies; this is often incorrect, and so terms should be disambiguated within the text before automatic category assignment is done. And finally, WordNet is not exhaustive and some mechanism is needed to improve coverage for unknown terms.

Acknowledgements Thanks to Lou Rosenfeld and Rashmi Sinha for their help finding participants, and to all the participants themselves. This work was funded in part by NSF DBI-0317510 and in part by the Summer Undergraduate Program in Engineering Research at Berkeley (SUPERB).

[5] MEdical Subject Headings, http://www.nlm.nih.gov/mesh/

References

Peter Anick and Susesh Tipirneni. 1999. The paraphrase search assistant:terminological feedback for iterative information seeking. In *Procs. of SIGIR'99*.

David Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *ACL '99*.

Douglas Cutting, David Karger D., Jan Pedersen, and John W. Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of SIGIR'92*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING '92*.

Thomas Hofmann. 1999. The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *Procs. of IJCAI'99*, Stolckholm, July.

Mika Kaki. 2005. Findex: Search result categories help users when document ranking fails. In *Proc. of CHI '05*.

Bernardo Magnini. 2000. Integrating subject field codes into WordNet. In *Procs. of LREC 2000*, Athens, Greece.

Rada Mihalcea and Dan I. Moldovan. 2001. Ez.wordnet: Principles for automatic generation of a coarse grained wordnet. In *Procs. of FLAIRS Conference 2001*, May.

Roberto Navigli, Paola Velardi, and Aldo Gangemi. 2003. Ontology learning and its application to automated terminology translation. *Intelligent Systems*, 18(1):22–31.

Craig Nevill-Manning, I. Witten, and G. Paynter. 1999. Lexically generated subject hierarchies for browsing large collections. *Inter. J. on Digital Libraries*, 2(2+3):111–123.

Wanda Pratt, Marti Hearst, and Larry Fagan. 1999. A knowledge-based approach to organizing retrieved documents. In *Procs. of AAAI 99*, Orlando, FL.

Louis Rosenfeld and Peter Morville. 2002. *Information Architecture for the World Wide Web: Designing Large-scale Web Sites*. O'Reilly & Associates, Inc.

Mark Sanderson and Bruce Croft. 1999. Deriving concept hierarchies from text. In *Procs. of SIGIR '99*.

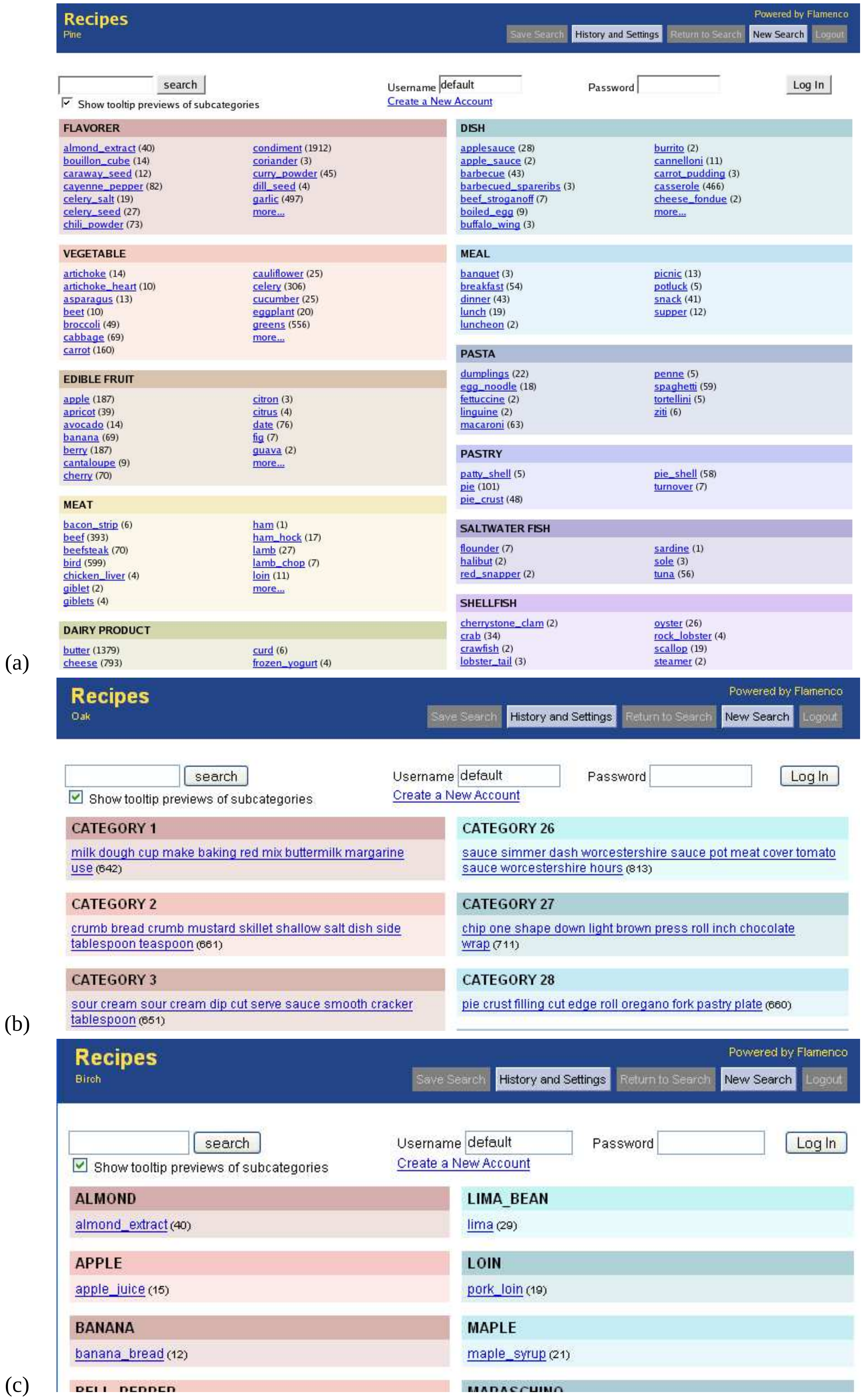
Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Emilia Stoica and Marti Hearst. 2004. Nearly-automated metadata hierarchy creation. In *Proc. of HLT-NAACL 2004*.

Piek Vossen. 2001. Extending, trimming and fussing wordnet for technical documents. In *NAACL 2001 Workshop and Other Lexical Resources*, East Stroudsburg, PA.

Dave Weinberger. 2005. Taxonomies and tags: From trees to piles of leaves. In *Release 1.0*, Feb.

Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. 2003. Faceted metadata for image search and browsing. In *Procs. of CHI '03*, Fort Lauderdale, FL, April.



FLAVORER

almond_extract (40)
bouillon_cube (14)
caraway_seed (12)
cayenne_pepper (82)
celery_salt (19)
celery_seed (27)
chili_powder (73)
condiment (1912)
coriander (3)
curry_powder (45)
dill_seed (4)
garlic (497)
more...

VEGETABLE

artichoke (14)
artichoke_heart (10)
asparagus (13)
beet (10)
broccoli (49)
cabbage (69)
carrot (160)
cauliflower (25)
celery (306)
cucumber (25)
eggplant (20)
greens (556)
more...

EDIBLE FRUIT

apple (187)
apricot (39)
avocado (14)
banana (69)
berry (187)
cantaloupe (9)
cherry (70)
citron (3)
citrus (4)
date (76)
fig (7)
guava (2)
more...

MEAT

bacon_strip (6)
beef (393)
beefsteak (70)
bird (599)
chicken_liver (4)
giblet (2)
giblets (4)
ham (1)
ham_hock (17)
lamb (27)
lamb_chop (7)
loin (11)
more...

DAIRY PRODUCT

butter (1379)
cheese (793)
curd (6)
frozen_yogurt (4)

DISH

applesauce (28)
apple_sauce (2)
barbecue (43)
barbecued_spareribs (3)
beef_stroganoff (7)
boiled_egg (9)
buffalo_wing (3)
burrito (2)
cannelloni (11)
carrot_pudding (3)
casserole (466)
cheese_fondue (2)
more...

MEAL

banquet (3)
breakfast (54)
dinner (43)
lunch (19)
luncheon (2)
picnic (13)
potluck (5)
snack (41)
supper (12)

PASTA

dumplings (22)
egg_noodle (18)
fettuccine (2)
linguine (2)
macaroni (63)
penne (5)
spaghetti (59)
tortellini (5)
ziti (6)

PASTRY

patty_shell (5)
pie (101)
pie_crust (48)
pie_shell (58)
turnover (7)

SALTWATER FISH

flounder (7)
halibut (2)
red_snapper (2)
sardine (1)
sole (3)
tuna (56)

SHELLFISH

cherrystone_clam (2)
crab (34)
crawfish (2)
lobster_tail (3)
oyster (26)
rock_lobster (4)
scallop (19)
steamer (2)

CATEGORY 1

milk dough cup make baking red mix buttermilk margarine use (642)

CATEGORY 2

crumb bread crumb mustard skillet shallow salt dish side tablespoon teaspoon (661)

CATEGORY 3

sour cream sour cream dip cut serve sauce smooth cracker tablespoon (651)

CATEGORY 26

sauce simmer dash worcestershire sauce pot meat cover tomato sauce worcestershire hours (813)

CATEGORY 27

chip one shape down light brown press roll inch chocolate wrap (711)

CATEGORY 28

pie crust filling cut edge roll oregano fork pastry plate (660)

ALMOND

almond_extract (40)

APPLE

apple_juice (15)

BANANA

banana_bread (12)

BELL PEPPER

LIMA_BEAN

lima (29)

LOIN

pork_loin (19)

MAPLE

maple_syrup (21)

MARASCHINO

Figure 5: Partial view of categories obtained by (a) Castanet, (b) LDA and (c) Subsumption on the Recipes collection, displayed in the Flamenco interface.