Human-competitive tagging using automatic keyphrase extraction

Olena Medelyan, Eibe Frank, Ian H. Witten
Computer Science Department
University of Waikato
{olena,eibe,ihw}@cs.waikato.ac.nz

Abstract

This paper connects two research areas: automatic tagging on the web and statistical keyphrase extraction. First, we analyze the quality of tags in a collaboratively created folksonomy using traditional evaluation techniques. Next, we demonstrate how documents can be tagged automatically with a state-of-the-art keyphrase extraction algorithm, and further improve performance in this new domain using a new algorithm, "Maui", that utilizes semantic information extracted from Wikipedia. Maui outperforms existing approaches and extracts tags that are competitive with those assigned by the best performing human taggers.

1 Introduction

Tagging is the process of labeling web resources based on their content. Each label, or *tag*, corresponds to a topic in a given document. Unlike metadata assigned by authors, or by professional indexers in libraries, tags are assigned by end-users for organizing and sharing information that is of interest to them. The organic system of tags assigned by all users of a given web platform is called a *folksonomy*.

In contrast to traditional taxonomies painstakingly constructed by experts, a user can add any tags to a folksonomy. This leads to the greatest downside of tagging, inconsistency, which originates in the synonymy and polysemy of human language, as well as in the varying degrees of specificity used by taggers (Golder and Huberman, 2006). In traditional libraries, *consistency* is the primary evaluation criterion of indexing (Rolling, 1981). Much work has been done on describing the statistical properties of folksonomies, such as tag distribution and co-occurrences (Halpin *et al.*, 2007; Sigurbjörnsson *et al.*, 2008; Sood *et al.*, 2007), but to our knowledge there has been none on assessing the actual quality of tags. How well do human taggers perform? How consistent are they with each other?

One potential solution to inconsistency in folksonomies is to use suggestion tools that automatically compute tags for new documents (e.g. Mishne, 2006; Sood *et al.*, 2007; Heymann *et al.*, 2008). Interestingly, the blooming research on automatic tagging has so far not been connected to work on keyphrase extraction (e.g. Frank *et al.*, 1999; Turney, 2003; Hulth, 2004), which can be used as a tool for the same task (note: we use *tag* and *keyphrase* as synonyms). Instead of simple heuristics based on term frequencies and co-occurrence of tags, keyphrase extraction methods apply machine learning to determine typical distributions of properties common to manually assigned phrases, and can include analysis of semantic relations between candidate tags (Turney, 2003). How well do state-of-the-art keyphrase extraction systems perform compared to simple tagging techniques? How consistent are they with human taggers? These are questions we address in this paper.

Until now, keyphrase extraction methods have primarily been evaluated using a single set of keyphrases for each document, thereby largely ignoring the subjective nature of the task. Collaboratively tagged documents, on the other hand, offer multiple tag assignments by independent users, a unique basis for evaluation that we capitalize upon in this paper.

The experiments reported in this paper fill these gaps in the research on automatic tagging and keyphrase extraction. First, we analyze tagging consistency on the *CiteULike.org* platform for organizing academic citations. Methods traditionally used for the evaluation of professional indexing will provide insight into the quality of this folksonomy. Next, we extract a high quality corpus from CiteULike, containing documents that have been tagged consistently by the best human taggers.

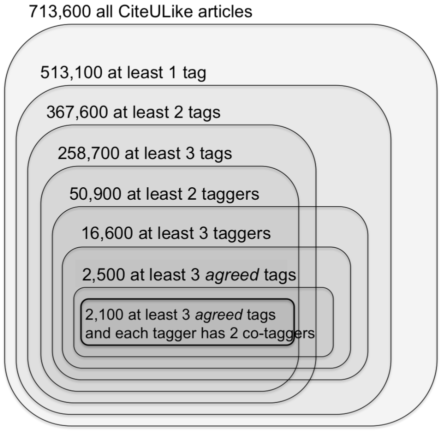
713,600 all CiteULike articles
513,100 at least 1 tag
367,600 at least 2 tags
258,700 at least 3 tags
50,900 at least 2 taggers
16,600 at least 3 taggers
2,500 at least 3 *agreed* tags
2,100 at least 3 *agreed* tags and each tagger has 2 co-taggers

Figure 1. Quality control of CiteULike data

Following that, our goal is to build a system that matches the performance of these taggers. We first apply an existing approach proposed by Brooks and Montanez (2006) and compare it to the keyphrase extraction algorithm Kea (Frank *et al.*, 1999). Next we create a new algorithm, called Maui, that enhances Kea's successful machine learning framework with semantic knowledge retrieved from Wikipedia, new features, and a new classification model. We evaluate Maui using tag sets assigned to the same documents by several users and show that it is as consistent with CiteULike users as they are with each other.

Most of the computation required for automatic tagging with this method can be performed offline. In practice, it can be used as a tag suggestion tool that provides users with tags describing the main topics of newly added documents, which can then be corrected or enhanced by personal tags if required. This will improve consistency in the folksonomy without compromising its flexibility.

2 Collaboratively-tagged Data

CiteULike.org is a bookmarking service that resembles the popular *del.icio.us*, but concentrates on scholarly papers. Rather than replicating the full text of tagged papers it simply points to them on the web (e.g. PubMed, CiteSeer, ScienceDirect, Amazon) or in journals (e.g. HighWire, Nature). This avoids violating copyright but means that the full text of articles is not necessarily available. When entering new resources, users are encouraged to assign tags describing their content or reflecting their own grouping of the information. However, the system does not suggest tags. Moreover, users do not see other users' tags and are thus not biased in their tag choices.

2.1 Extracting a high quality tagged corpus

The CiteULike data set is freely available and contains information about which documents were tagged with what tags by which users (although identities are not provided).

CiteULike's 22,300 users have tagged 713,600 documents with 2.4M "tag assignments"— single applications of a tag by a user to a document. The two most popular tags, *bibtex-import* and *no-tag*, indicate an information source and a missing tag respectively. Most of the remainder describe particular concepts relevant to the documents. We exclude non-content tags from our experiments, e.g. personal tags like *to-read* or *todo*. Note that spam entries have been eliminated from the data set.

Because CiteULike taggers are not professional indexers, high quality of the assigned topics cannot be guaranteed. In fact, manual assessment of users' tags by human evaluators shows precision of 59% (Mishne, 2006) and 49% (Sood *et al.*, 2006). However, why is the opinion of human evaluators valued more than the opinion of taggers? We propose an alternative way of determining ground truth using an automatic approach to determine reliable tags: We concentrate on a subset of CiteULike containing documents that have been indexed with at least three tags on which at least two users have agreed.

In order to be able to measure the tagging consistency between the users, and then compare it to the algorithm's consistency, we need taggers who have tagged documents that some others had tagged. We say that two users are "co-taggers" if they have both tagged at least one common document. As well as restricting the document set, we only include taggers who have at least two co-taggers.

Figure 1 shows the proportions of CiteULike documents that are discarded in order to produce our high quality data set. The final set contains only 2,100 documents (0.3% of the original). Unfortunately, many of these are unavailable for download—for example, books at Amazon.com and ArXiv.org references cannot be crawled. We further restrict attention to two sources: HighWire and Nature, both of which provide easily-accessible PDFs of the full text.

The result is a set of 180 documents indexed by 332 taggers. A total of 4,638 tags were assigned by all taggers to documents in this set; however, the number of tags on which at least two users agreed is significantly smaller, namely 946. Still, this results in accurate tag sets that contain an average of five tags per document.

tagger	co-taggers	documents	consistency
1	1	5	71.4
2	1	5	71.4
3	6	5	57.9
4	6	6	51.0
5	11	12	50.4
6	2	5	50.1
7	4	6	48.3
8	8	8	47.1
9	13	16	45.4
10	12	8	44.4
11	7	6	43.5
12	7	6	41.7
13	8	5	40.9
14	7	6	39.7
15	9	13	38.8
16	4	5	38.4
17	12	9	37.3
18	4	14	36.1
19	9	8	35.9
20	10	11	33.7
21	7	6	33.1
22	6	5	33.0
23	7	10	32.1
24	11	16	31.7
25	8	13	30.6
26	6	8	30.6
27	9	6	29.8
28	10	12	29.0
29	8	6	28.8
30	9	10	27.9
31	10	8	26.7
32	8	7	26.3
33	10	5	25.6
34	8	7	21.0
35	9	9	18.3
36	3	6	7.9
average	**7.5**	**8.1**	**37.7**

Table 1. Consistency of the most prolific and most consistent taggers

Note that traditionally much smaller data sets are used to assess consistency of human indexers, because such sets need to be created specifically for the experiment. Collaborative tagging platforms like CiteULike can be mined for large collections of this kind in natural settings.

Most documents in the extracted set relate to the area of bioinformatics. To give an example, a document entitled *Initial sequencing and comparative analysis of the mouse genome* was tagged by eight users with a total of 22 tags. Four of them agreed on the tag *mouse*, but one used the broader term *rodents*. Three agreed on the tag *genome*, but one added *genome paper*, and another used the more specific *comparative genomics*. There are also cases when tags are written together, e.g. *genomepaper*, or with a prefix *key genome*, or in a different grammatical form: *sequence* vs. *sequencing*. This example shows that many inconsistencies in tags are not caused by personalized tag choices as Chirita *et al.* (2007) suggest, but rather stem from the lack of guidelines and uniform tag suggestions that a bookmarking service could provide.

2.2 Measuring tagging consistency

Traditional indexers aim for consistency, on the basis that this will enhance document retrieval (Leonard, 1975). Consistency is measured using experiments in which several people index the same documents—usually a small set of a few dozen documents. It is computed for pairs of indexers, by formulae such as Rolling's (1981):

$$Consistency(I_1,I_2) = \frac{2C}{A+B},$$

where C is the number of tags (index terms) indexers I_1 and I_2 have in common and A and B is the size of their tag sets respectively.

In our experiments, before computing the number of terms in common, we stem each tag with the Porter (1980) stemmer. For example, the overlap C between the tag sets {*complex systems, network, small world*} and {*theoretical, small world, networks, dynamics*} consist of the two tags {*network, small world*}, and the consistency is 2×2/(3+4) = 0.57.

To compute the overall consistency of a particular indexer, this figure is averaged over all documents and co-indexers. There were no cases where the same user reassigned tags to the same articles, so computing intra-tagger consistency, although interesting, was not impossible.

To our knowledge, traditional indexing consistency metrics have not yet been applied to collaboratively tagged data. However, experiments on determining tagging quality do follow the same idea. For example, Xu *et al.* (2006) define an authority metric that assigns high scores to those users who match other users' choices on the same documents, in order to eliminate spammers.

2.3 Consistency of CiteULike taggers

In the collection of 180 documents tagged by 332 users described in Section 3.1, each tagger has 18 co-taggers on average, ranging from 2 to 129, and has indexed 1 to 25 documents. For each user we compute the consistency with all other users who tagged the same document. Consistency is then averaged across documents. We found that the distribution of per-user consistency resembles a power law with a few users achieving high consistency values and a long tail of inconsistent taggers. The maximum consistency in this group is 92.3% and the average is 18.5%. The average consistency of the most prolific 70 indexers—those who have indexed at least five documents—is in the same range, namely 18.4%. The consistency of traditional approaches to free indexing is reported to be between 4% and 67%, with an average of 27% depending on what aids are used (Leininger, 2000).

It is instructive to consider the group of best taggers. We define these as the ones who (a) exhibit greater than average consistency with all others, and (b) are sufficiently prolific, i.e. have tagged at least five documents. There are 36 such taggers; Table 1 lists their consistency within this group. The average consistency they achieve as a group is 37.7%, which is the similar to the average consistency of professionals (Leininger, 2000).

The above consistency analysis provides insight into the tagging quality of the best CiteULike users, based on HighWire and Nature articles. For the purposes of this paper, it shows how the tagging community can be restricted to a best-performing group of taggers by measuring their consistency. This is helpful for testing the performance of automatic tagging (Section 4.4).

3 Automatic tagging with Maui

Maui is a general algorithm for automatic topical indexing based on the Kea system (Frank *et al.*, 1999).[1] It works in two stages: candidate selection and machine learning based filtering. In this paper, we apply it to automatic tagging. In the candidate selection stage, Maui first determines textual sequences defined by orthographic boundaries and splits these sequences into tokens. Then all n-grams up to a maximum length of 3 words that do not begin or end with a stopword are extracted as candidate tags. To reduce the number of candidates, all those that appear only once are discarded. This speeds up the training and the extraction process without impacting the results. In the filtering stage several features are computed for each candidate, which are then input to a machine learning model to obtain the probability that the candidate is indeed a tag.

Maui's architecture resembles that of many other supervised keyphrase extraction systems (Turney, 2000; Hulth 2004; Medelyan *et al.*, 2008). However, this architecture has not previously been applied to the task of automatic tagging.

3.1 Features indicating significance

We now describe the features used in the classification model to determine whether a phrase is likely to be a tag. We begin with three baseline features used in Kea (Frank *et al.*, 1999), and extend the set with three features that have been found useful in previous work. We also add three new features that have not been evaluated before: *spread*, *semantic relatedness* and inverse *Wikipedia linkage*. All Wikipedia-based features are computed using the WikipediaMiner toolkit.[2]

1. TF×IDF combines the frequency of a phrase in a particular document with its inverse occurrence frequency in general use (Salton and McGill, 1983). This score is high for rare phrases that appear frequently in a document and therefore are more likely to be significant.

2. Position of the first occurrence is computed as the relative distance of the first occurrence of the candidate tag from the beginning of the document. Candidates with very high or very low values are likely to be tags, because they appear either in the opening document parts such as title, abstract, table of contents, and introduction, or in the document's final sections such as conclusion and reference lists.

3. Keyphraseness quantifies how often a candidate phrase appears as a tag in the training corpus. Automatic tagging approaches utilize the same information: Mishne (2006) and Sood *et al.* (2006) automatically suggest tags previously assigned to similar documents. However, in Maui (as in Kea) this feature is just one component of the overall model. Thus if a candidate never appears as a keyphrase in the training corpus, it can still be extracted if its other feature values are significant enough.

4. Phrase length is measured in words. Generally speaking, the longer the phrase, the more specific it is. Training captures and quantifies the specificity preference in a given training corpus.

5. Node degree quantifies the semantic relatedness of a candidate tag to other candidates. Turney (2003) computes semantic relatedness using search engine statistics. Instead, following Medelyan *et al.* (2008), we utilize Wikipedia hyperlinks for this task. We first map each candidate phrase to its most common Wikipedia page. For example, the word *Jaguar* appears as a link anchor in Wikipedia 927 times. In 466 cases it links to the article *Jaguar cars*, thus the commonness of this mapping is 0.5. In 203 cases it links to the animal description, a commonness of

[1] Maui is open-source and available for download at *http://maui-indexer.googlecode.com*

[2] http://wikipedia-miner.sourceforge.net/

0.22. We compute the node degree of the corresponding Wikipedia article as the number of hyperlinks that connect it to other Wikipedia pages that have been identified for other candidate tags from the same document. A document that describes a particular topic will cover many related concepts, so high node degree—which indicates strong connectivity to other phrases in the same document—means that a candidate is more likely to be significant.

6. Wikipedia-based keyphraseness is the likelihood of a phrase being a link in the Wikipedia corpus. It divides the number of Wikipedia pages in which the phrase appears in the anchor text of a link by the total number of Wikipedia pages containing it. We multiply this number by the phrase's document frequency.

The new features proposed in this paper are the following:

7. Spread of a phrase is the distance between its first and last occurrences in a document. Both values are computed relative to the length of the document (see feature 2). High values help to determine phrases that are mentioned both in the beginning and at the end of a document.

8. Semantic relatedness of a phrase has already been captured as the node degree (see feature 5). However, recent research allows us to compute semantic relatedness with better techniques than mere hyperlink counts. Milne and Witten (2008) propose an efficient Wikipedia based approach that is nearly as accurate as human subjects at quantifying the relationship between two given concepts. Given a set of candidate phrases we determine the most likely Wikipedia articles that describe them (as explained in feature 5), and then determine the total relatedness of a given phrase to all other candidates. The higher the value, the more likely is the phrase to be a tag.

9. Inverse Wikipedia linkage is another feature that utilizes Wikipedia as a source of language usage statistics. Here, again given the most likely Wikipedia article for a given phrase, we count the number of other Wikipedia articles that link to it and normalize this value as in inverse document frequency:

$$IWL = -\log_2 \frac{linksTo(A_P)}{N}$$

where $linksTo(A_P)$ is the number of incoming links to the article A representing the candidate phrase P, and N is the total number of links in our Wikipedia snapshot (52M). This feature highlights those phrases that refer to concepts commonly used to describe other concepts.

3.2 Machine learning in Maui

In order to build the model, we use the subset of the CiteULike collection described in Section 3.1. For each document we know a set of tags that at least two users have agreed on. This is used as ground truth for building the model. For each training document, candidate phrases (i.e. n-grams) are identified and their feature values are calculated as described above.

Each candidate is then marked as a positive or negative example, depending on whether users have assigned it as a tag to the corresponding document. The machine-learning model is constructed automatically from these labeled training examples using the WEKA machine learning workbench. Kea (Frank *et al.*, 1999) uses the Naïve Bayes classifier, which implicitly assumes that the features are independent of each other given the classification. However, Kea uses only two or three features, whereas Maui combines nine features amongst which there are many obvious relationships, e.g. first occurrence and spread, or node degree and semantic relatedness. Consequently, we also consider bagged decision trees, which can model attribute interactions and do not require parameter tuning to yield good results. Bagging learns an ensemble of classifiers and uses them in combination, thereby often achieving significantly better results than the individual classifiers (Breiman, 1996). Different trees are generated by sampling from the original dataset with replacement. Like Naïve Bayes, bagged trees yield probability estimates that can be used to rank candidates.

To select tags from a new document, Maui determines candidate phrases and their feature values, and then applies the classifier built during training. This classifier determines the probability that a candidate is a tag based on relative frequencies observed from the training data.

4 Evaluation

Here we describe the data used in the experiments and the results obtained, addressing the following questions:

1. How does a state-of-the-art keyphrase extraction method perform on collaboratively tagged data, compared to a baseline automatic tagging method?
2. What is the performance of Maui with old and new features?
3. How consistent are Maui's tags compared to those assigned by human taggers?

		P	R	F
1	Top words based on TF×IDF	16.8	17.3	17.0
2	Top phrases based on TF×IDF	14.4	16.0	15.2
3	Kea (TF×IDF, 1st occur)	20.4	22.3	21.3
4	Kea (+keyphraseness)	**41.1**	**43.1**	**42.1**

Table 2. Baseline auto-tagging approach vs. Kea

4.1 Evaluation method

The evaluation was performed using a set of 180 documents, described in Section 3.1, each tagged with at least three tags on which two users have agreed. In the following, unless explicitly stated otherwise, these are the only tags we use. We consider them to be ground truth. There are on average five such tags per document, and our goal is to extract tag sets that contain them all.

We regard a predicted tag as "correct" if it matches one of the ground truth tags after using the Porter stemmer. We measure performance by computing Precision (the percentage of correct extracted tags out of all extracted), Recall (the percentage of correct extracted tags out of all correct) and F-Measure (the harmonic mean of the two). Given the set {*yeast* (4), *network* (3), *regulation* (2), *metabolic* (2)} of ground truth tags, where the numbers in parenthesis show how many users have assigned each one, and the set {*network*, *metabolic*, *regulatory*, *ChIP-chip*, *transcription*} of predicted tags, three out of five predicted terms are correct, yielding a precision of 60%, and three out of four ground-truth terms are extracted, a recall of 75%. The F-measure combining the two values is 67%.

The reported precision and recall values are averaged over all test documents. We use 10-fold cross-validation for evaluation, which allows us to use all 180 documents as test documents without introducing optimistic bias in the performance measures obtained.

The results obtained in Sections 4.2 and 4.3 using this evaluation provide answers to the first two questions above. To answer the third we compare the indexing consistency of Maui to that of CiteULike users in Section 4.4. Here, we consider the assigned tag sets individually and compute the consistency of Maui with each tagger as described in Section 3.2. We compare Maui both to all 332 users who tagged these documents, and to the 36 best taggers identified in Section 3.3.

4.2 Keyphrase extraction vs. auto-tagging

As noted earlier, Brooks and Montanez (2006) automatically determine tags by extracting terms with the highest TF×IDF values for each post and argue that their quality is perhaps better than that of manual tags. Note that they only use one-word tags. We evaluate this approach using our 180 test documents and cross-validation, and compare the top five extracted tags with those assigned manually. Comparing the first two rows of Table 2 shows that using multi-word phrases as candidate tags (Section 4) is less accurate than using single words, which gives an overall F-Measure of 17%. Multi-words have higher TF×IDF values, but single words are the majority among the users' tags. The length feature applied in the next section helps to capture this characteristic, without compromising Maui's ability to assign correct multi-words tags.

Adding a second feature, the position of the first occurrence, and using Kea's Naïve Bayes model to learn their conditional distribution, improves the results by 5 percentage points (row 3). Adding the keyphraseness feature (row 4) nearly doubles the F-Measure, from 21.3 to 42.1%. This shows that CiteULike users tend to re-assign existing tags.

		P	R	F
1	TFxIDF	14.4	16	15.2
2	1st occurrence	5.4	5.4	5.4
3	Keyphraseness	25.2	26.3	25.5
4	Length	2.1	2.1	2.1
5	Node degree	8.3	9.0	8.6
6	Wikipedia keyphraseness	16.9	18.3	17.6
7	Spread	12.1	13.0	12.5
8	Semantic relatedness	7.1	7.3	7.2
9	Inverse Wikipedia linkage	7.3	6.8	7.0

Table 3. Evaluation of individual features

4.3 Maui with additional features

To evaluate Maui let us first consider the individual performance of old and new features, as shown in Table 3. Rows 1 to 3 evaluate the standard features used by Frank *et al.* (1999); Rows 4 to 6 evaluate features that were previously used in Kea for controlled indexing (Medelyan *et al.*, 2006) and which we have adapted in Maui for free indexing. Rows 7 to 9 evaluate the three new features of Maui. The values can be compared to keyphrase extraction by chance (F-Measure = 1%) and to the multi-word TF×IDF baseline in Table 2, row 2 (F-Measure = 15.2%). The strength of these features varies from 2.1 to 25.5% (F-Measure). The strongest ones are keyphraseness, Wikipedia keyphraseness, TF×IDF and spread.

Table 4 demonstrates Maui's performance when the features are combined and shows how the two different classifiers, Naïve Bayes (left) and bagged decision trees (right), compare to

		Naïve Bayes			Bagged decision trees		
		P	R	F	P	R	F
1	Features 1 – 3	**41.1**	**43.1**	**42.1**	40.3	42.2	41.2
2	Features 1 – 6	38.9	41.1	40.0	40.3	42.6	41.4
3	Features 1 – 3, 7 – 9	39.3	41.1	40.2	43.7	46.2	44.9
4	Features 1 – 9	37.6	39.6	38.6	**45.7**	**48.6**	**47.1**

Table 4. Combining all features in Maui

each other. The baseline in row 1 (left) shows Kea's performance, using TF×IDF, first occurrence, keyphraseness and Naïve Bayes to combine them (same as row 4 in Table 2). Using decision trees with these three features does not improve the performance (row 1, right). The following row combines the three original features with length, node degree and Wikipedia-based keyphraseness. In contrast to previous research (Medelyan *et al.*, 2008), in this setting we do not observe an improvement with either Naïve Bayes or bagged decision trees. In row 3 we combine the three original features with the three new ones introduced in this work. While Naïve Bayes' values are lower than the baseline, with bagged decision trees Maui's F-Measure improves from 41.2 to 44.9%. The best results are obtained by combining all nine features, again using bagged decision trees, giving in row 4 (right) a notably improved F-Measure of 47.1%. The recall of 48.6% shows that we match nearly half of all tags on which at least two human taggers have agreed.

Given this best combination of features, we eliminate each feature one by one starting from the individually weakest feature, in order to determine the contribution of each feature to this overall result. Table 5 compares the values and only bagged decision trees are used this time. The 'Difference' column quantifies the difference between the best F-Measure achieved with all 9 features and excluding the one that is examined in that row. Interestingly, one of the strongest features, TF×IDF, is the one that contributes the least when all features are combined, while the contribution of the strongest feature—keyphraseness—is, as expected, the highest, adding 16.9 points. The second most important feature is Wikipedia keyphraseness, contributing 4 percentage points to the overall result.

Features	F-Measure	Difference
All 9 Features	47.1	
– Length	45	2.1
– 1st occurrence	45.6	1.5
– Inverse Wikip linkage	45.1	2
– Semantic relatedness	45.4	1.7
– Node degree	46	1.1
– Spread	46.4	0.7
– TFxIDF	46.8	0.3
– Wikip keyphraseness	43.1	4
– Keyphraseness	30.2	16.9
Non-Wikip features	41.7	5.4

Table 5. Evaluation using feature elimination

Since some of the features in the best performing combination rely on Wikipedia as a knowledge source, it is interesting to determine Wikipedia's exact contribution. The last row of Table 5 combines the following features: TF×IDF, first occurrence, keyphraseness, length and spread. The F-Measure is 5.4 points lower than that of Maui with all 9 features combined. Therefore, the contribution of Wikipedia-based features is significant.

4.4 Maui's consistency with human taggers

In Section 2.3 we discussed the indexing consistency of CiteULike users on our data. There are a total of 332 taggers and their consistency with each other is 18.5%. Now, we use results obtained with Maui during the cross-validation, when all 9 features and bagged decision trees are used (Table 4, row 4, right; see examples in Table 5), and compute how consistent Maui is with each human user, based on whatever document this user has tagged. Then we average the results to obtain the overall consistency with all 332 users.

Maui's consistency with the 332 human taggers ranges from 0 to 80%, with an average of 23.8%. The only cases where very low consistency was achieved are those where the human has only assigned a few tags per document (one to three), or has some idiosyncratic tagging behavior (for example, one tagger adds the word *key* in front of most tags). Still, with an average of 23.8%, Maui's performance is over 5 points higher than that of an average CiteULike tagger (18.5%)—and note this group only includes taggers who have at least two co-taggers.

In Section 2.3 we were also able to determine a smaller group of users who perform best and are most prolific. This group consists of 36 taggers whose consistency exceeds the average of the original 332 users. These 36 taggers have tagged a total of 143 documents with an average consistency of 37.6%. Maui's consistency with

Document	86865. Neural correlates of decision variables in parietal cortex. Platt and Glimcher. *Nature* 400,15 (1999)	44. Exploring complex networks. Strogatz. *Nature* 410, 8 (2001)	353537. Computational roles for dopamine in behavioural control. Montague et al. *Nature* 431, 14 (2004)	101. Network motifs: simple building blocks of complex networks. Milo et al. *Science* 298, 824 (2002)
Tags assigned by CiteULike taggers	**decision making** **decisionmaking** **lip** **monkey** **neurophysiology** **reward** ***Idiosyncratic:*** brain, choice, cortex, decision, electrophysiology, eye-movements, limitations, monkeys, neuroeconomics, neurons, neuroscience, other, ppc, quals, reinforcementlearning	**complex** **complexity** **complex networks** **graph** **networks** **review** **small world** **social networks** **survey** ***Idiosyncratic:*** 2001, adaptive systems, bistability, coupled oscillator, graph mining, graphs, explorig, network biological, neurons, strogatz	**dopamine** **neuroscience** **reinforcement learning** **review** ***Idiosyncratic:*** action selection, attention, behavior, behavioral control, cognitive control, learning, network, reinforcementlearning, reward, td model	**applied math** **combinatorics** **complexity** **motifs** **network** **original** **sub graph pattern** ***Idiosyncratic:*** 2002, datamining, data mining, graphs, link analysis, modularity, net paper, patterns, protein, science, sysbio, web characterization, web graph
Tags assigned by Maui	**cortex** **decision** **lip** **monkey** visual	**complex networks** **networks** **review** synchronization **graph**	**dopamine** **learning** **neuroscience** **review** **reward**	complex networks **network** **motifs** gene **complex**

Table 6. Tags assigned by CiteULike taggers and Maui to four sample documents

these taggers ranges from 11.5% to 56%, with an average of 35%. This places it only 2.6 percentage points behind the average performance of the best CiteULike taggers. In fact, it outperforms 17 of them (cf. Table 1).

4.5 Examples

Table 6 compares Maui with some of CiteULike's best human taggers on four randomly chosen test documents. Boldface in the taggers' row indicates a tag that has been chosen by at least two other human taggers; the remaining tags have been chosen by just one human. Boldface in Maui's row shows tags that match human tags. For each document Maui extracts several tags assigned by at least two humans. The other tags it chooses are generally chosen by at least one human tagger, and even if not, they are still related to the main theme of the document.

5 Discussion and related work

It is possible to indirectly compare the results of several previously published automatic tagging approaches with Maui's. For each paper, we compute Maui's results in settings closest to the reported ones.

Brooks and Montanez (2006) extract terms with the highest TF×IDF values as tags for posts on *technorati.com*. They do not report precision and recall values for their system, but our re-implementation resulted in precision of 16.8% and recall of 17.3% for the top five assigned tags, compared to those agreed to by at least two CiteULike users on 180 documents. Adding eight additional features and combining them using machine learning gives a clear improvement—Maui achieves 45.7% and 48.7% precision and recall respectively.

Mishne (2006) uses TF×IDF-weighted terms as full-text queries to retrieve posts similar to the one being analyzed. Tags assigned to these posts are analyzed to retrieve the best ones using clustering and heuristic ranking; tags assigned by the given user receive extra weight. Mishne performs manual evaluation on 30 short articles and reports precision and recall for the top ten tags of 38% and 47% respectively. We matched Maui's top ten terms to all tags assigned to 180 documents automatically and obtained precision and recall of 44% and 29% respectively. (We believe that manual rather than automatic evaluation would be likely to give a far more favorable assessment of our system.)

Chirita *et al.* (2007) aim to extract personalized tags. Given a web page, they first retrieve similar documents stored on the user's desktop and then determine keywords for these documents. They evaluate different term scoring techniques, such as term and document frequency, lexical dispersion, sentence scoring, and term co-occurrence. Like the Kea algorithm, the best formula combines term frequency with the position of the first occurrence of the term, normalized by page length. It yields a precision of 80% for the top four tags assigned to 30 large websites (32Kbytes), again evaluated manually. Our documents are considerably longer (47Kbytes) and thus more difficult to work with, nevertheless Maui achieves only slightly lower values, from 66% to 80%, when evaluating automatically against user-assigned tags. (The above caveat regarding automatic and manual assessment applies here too.)

Budura *et al.* (2008) develop a scoring formula that combines three features (tag frequency, tag co-occurrence and document similarity) and manually evaluate it on ten CiteULike documents. Their precision for the top three to five tags ranges from 66% to 77%, slightly worse than in our paper (66% to 80%).

The only reported automatic evaluation of tags was found in Sood *et al.* (2006), where TagAssist was tested on 1000 blog posts. This algorithm is similar to Mishne's (2006), but uses centroid-based clustering. Exact matching of TagAssist's tags against existing ones yielded precision and recall of 13.1% and 22.8% respectively. This is substantially lower than Maui's 45.75% and 48.7% obtained with best settings (Section 4.3).

Note that this indirect comparison does not reveal the true ranking of approaches, because their task definitions and test sets are slightly different. It would be interesting to compare other systems on the multiple tagger set described in this paper, as we believe this would more objectively reflect the performance of humans and algorithms.

6 Conclusions

This paper has introduced a systematic way of evaluating automatic tagging techniques without the need for manual inspection. We have shown how documents with multiple tag sets can be used in conjunction with a standard consistency measure to identify a robust test corpus for these techniques. Based on the evaluation methodology developed, we have shown that machine-learning-based automatic keyphrase extraction produces tag sets that exhibit consistency on a par with that achieved by the best human taggers. Our results also show a substantial improvement on an existing automatic tagging approach based on TF×IDF, and the results compare well to other systems.

The success of automatic keyphrase extraction depends primarily on the quality of the features that are provided to the machine learning algorithm involved. In this paper we have evaluated nine different features, including two novel Wikipedia-based semantic features, and found that their combination used in conjunction with bagged decision trees produces the best performance.

References

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2): 123–140.

Brooks, C. H. and N. Montanez. 2006. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proc. Int. Conf. on World Wide Web,* Edinburgh, UK. pp. 625–632. New York, NY, USA. ACM Press.

Budura, A., S. Michel, P. Cudre-Mauroux, and K. Aberer. 2008. To tag or not to tag - harvesting adjacent metadata in large-scale tagging systems. In *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Singapore. pp. 733–734. New York, NY, USA: ACM Press.

Chirita, P. A., S. Costache, W. Nejdl, and S. Handschuh. 2007. P-tag: large scale automatic generation of personalized annotation tags for the web. In *Proc. Int. Conf. on World Wide Web,* Banff, Canada. pp. 845–854. New York, NY, USA: ACM Press.

Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden. pp. 668–673. San Francisco, CA: Morgan Kaufmann Publishers.

Golder, S. A. and B. A. Huberman. 2006. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2): 198–208.

Halpin, H., V. Robu, and H. Shepherd. 2007. The complex dynamics of collaborative tagging. In *Proc. Int. Conf. on World Wide Web,* pp. 211–220. New York, NY, USA: ACM Press.

Heymann, P., D. Ramage, and H. Garcia-Molina. 2008. Social tag prediction. In *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Singapore. pp. 531–538. New York, NY, USA: ACM Press.

Hulth, A. 2004. *Combining machine learning and natural language processing for automatic keyword extraction.* Ph.D. thesis, Dep. of Computer and Systems Sciences, Stockholm University.

Leonard, L. E. 1975. *Inter-indexer consistency and retrieval effectiveness: measurement of relationships.* Ph.D. thesis, Grad. School of Library Science, Univ. of Illinois, Urbana-Champaign, IL.

Leininger, K. 2000. Interindexer consistency in PsycInfo. *Journal of Librarianship and Information Science* 32(1): 4–8.

Medelyan, O., I. H. Witten and D. Milne. 2008. Topic indexing with Wikipedia. In *Proc. of AAAI'08 Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, Chicago, USA. pp. 19–24.

Mishne, G. 2006. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proc. Int. Conf. on World Wide Web*, Edinburgh, UK. pp. 953–954. New York, NY, USA. ACM Press

Porter, M. F. 1980. An algorithm for suffix stripping, *Program*, 14(3): 130−137.

Rolling, L. 1981. Indexing Consistency, Quality and Efficiency. *Information Processing & Management* 17(2): 69–76.

Salton, G. and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill New York.

Sigurbjörnsson, B. and R. van Zwol. 2008. Flickr tag recommendation based on collective knowledge. In *Proc. Int. Conf. on World Wide Web,* Beijing, China. pp. 327–336. New York, NY, USA: ACM Press.

Sood, S., K. Hammond, S. Owsley, and L. Birnbaum. 2007. TagAssist: Automatic tag suggestion for blog posts. of *Int. Conf. on Weblogs and Social Media*, Boulder, Colorado. Menlo Park, CA.

Turney, P. D. 2003. Coherent keyphrase extraction via web mining. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence,* Acapulco, Mexico. pp. 434–439. San Francisco, CA: Morgan Kaufmann Publishers.

Xu, Z., Fu, Y., Mao, J., and D. Su. 2006. Towards the Semantic Web: Collaborative tag suggestions. In *Proc. Collaborative Web Tagging Workshop at the Int. Joint Conf. on Artificial Intelligence,* Stockholm, Sweden.