

Performance tags – who's running the show?

Emma Tonkin ¹, Gregory J. L. Tourte ² and Alla Zollers ³

¹ e.tonkin@ukoln.ac.uk

UKOLN, University of Bath, UK

² g.j.l.tourte@bristol.ac.uk

University of Bristol, UK

³ alla.zollers@ucl.edu

University of California, Los Angeles

Abstract

We describe a pilot study which specifically examines the prevalence and characteristics of performance tags on several sites. Identifying post-coordination of tags as a useful step in the study of this phenomenon, as well as other approaches to leveraging tags based on text and/or sentiment analysis, we demonstrate an approach to automation of this process, postcoordinating (segmenting) terms by means of a probabilistic model based around Markov chains. The effectiveness of this approach to parsing is evaluated with respect to the wide range of constructions visible on various services. Several candidate approaches for the latter stages of automated classification are identified.

Introduction

Yeung et al (2007) describes social tagging within a social network as a tripartite graph of user, tag and resource, describing the effect of this linking as 'mutual contextualization'. In this view, semantics constitute socially shared constructions that alter as the network evolves and are acquired through association with other elements. It follows that, beyond classification of the object through association with the tag and definition of the tag via a link to the object, the network can equally be seen as a means of publishing assertions about oneself. In other words, linking oneself to a resource by means of an expression gives rise to the possibility that others will see these links as a source of information about their author. Such a use case connects classification to microblogging, and may be applied for information exchange between colleagues or user groups, as a means of persuasion, performance, or developing a public identity or online profile (Zollers, 2007; Tonkin, 2008), using various strategies to situate the written word within a socio-cultural context.

In addition to the application of tags to organise content for later retrieval, tags are often employed to convey information beyond their primary use as symbols representing the theme or content of an object, containing keywords, interpretative data, reactions, and functional/action tags (Golder & Huberman, 2006; Kipp, 2007). Given the nature of some of the tags such as “waste of time and money” when referring to movie, or “makes me wish for the sweet release of death” when describing a musical CD, it is reasonable to hypothesise that people are also utilizing tags to communicate with other members of the site. When people utilize tags for communication, the users may perceive or construct an intended audience for the tags.

One specialized way in which people communicate through tags is by creating performance tags, which are very unique and creative. Zollers (2007) coined the term 'performance tags' to describe particular forms of tag, those which suggest that the tags are in some sense authored as part of a performance, played on behalf of a real or notional audience. She quotes Schechner (2001) in saying that in everyday life “to perform is to show off, to go to extremes, to underline an action to those who are watching.”

Tags such as “makes me wish for the sweet release of death” can be categorized as performance tags. Such a tag has an informational dimension – it suggests that the resource to which it has been applied is better avoided. The second informational dimension refers to the author themselves, by means of the provision of a dramatised reaction to the resource, much in the same way as an individual's everyday use of vocabulary and register provides clues as to identity.

Intuitively, one might not expect to see as many terms with negative affect as positive, since there would seem to be no reason to involve oneself with a resource that one does not like or consider useful. One might expect the most interest to be taken by fans of the work in question. The social phenomenon of the 'fan' is well-known, usually inspiring mental images of nerdy groups of people who meet online and in person to share their interests, and make use of all sorts of topical resources – and imagery in constructing a publicly visible identity that

proclaims their interests.

The phenomenon of the anti-fan, though perhaps less well-known, is society's answer to this. Anti-fans, according to Gray (2003) are those who strongly dislike a given text or genre, considering it inane, stupid, morally bankrupt and/or aesthetic drivel... variously bothered, insulted or otherwise assaulted by [the presence of a given text or resource]. They too meet, share and develop their opinions and publicise them in some cases widely. Professing enjoyment, interest, hatred or disgust towards a resource can all serve as polarising factors in a social situation. Gray points out that the resource may serve as a symbol through which to express a political viewpoint, so that there is a strong correlation between 'loving or disliking The Simpsons and seeing it, respectively, as critical of America and American life, or as yet another symbol of crass American cultural chauvinism.'

Returning to the theme of opinion as performance, we note that performance tags are very visible on certain sites. However, by observation, they do not seem to be a feature of all tagging systems. In the first part of this paper, we examine a number of tagging systems manually in a small pilot study, with the aim of ascertaining the actual frequency with which these arise. The latter part of this paper documents an approach to automating the discovery of performance tags, notably the problem of segmentation of a body of tags into 'phrases' – that is, post-coordination. This work exists in contrast to the research reported by Tonkin (2006), in which the problem of untangling a specific form of precoordinated tags (compounds) is examined, and a candidate solution is described.

Study 1: A pilot study aiming to characterise performance tags across sites

We describe a pilot study which specifically examines the prevalence and characteristics of performance tags on the following sites: Amazon, CiteULike, Connotea, Del.icio.us, last.fm, Panoramio, Slashdot and YouTube. This information will enable us to compare and contrast the tagging behaviors exhibited across various sites, as well as gain a deeper understanding of the characteristics of performance tags. Our hypothesis is that they are most commonly applied on sites that deal with popular culture, such as music, movies and hobbies.

Methodology

To this end, a randomised sample of tags are taken from each site. The precise means by which tags are gathered and randomised is dependent on the available interfaces and structure of each site, ranging from the use of provided sample data to data extracted via a purpose-built web spider. Component tags from each sample are manually segmented/post-coordinated (Catarino & Baptista, 2008) and classified according to several metrics; tag length in words and characters, tag structure according to part-of-speech tagging of component elements, and the status of each tag as an example of a performance tag.

Discussion of results

Manually counted results show considerable variation in the use of tags in general, both within and between systems. Occurrence of compounds (phrases, on systems that allow multiword terms) ranges from 15% to 50%. Noun phrases appear more commonly on academic sites than on popular culture sites; the converse is true of adjectives. Occurrence of performance tags range from 0% in the case of Panoramio to 47% in the case of Slashdot, with a distribution that concurs with the presented hypothesis. Tag syntax is a good although fallible predictor of expressivity as a performance tag, as are simple heuristics such as counts of tag length/number of words. The results are summarised in Figure 1.

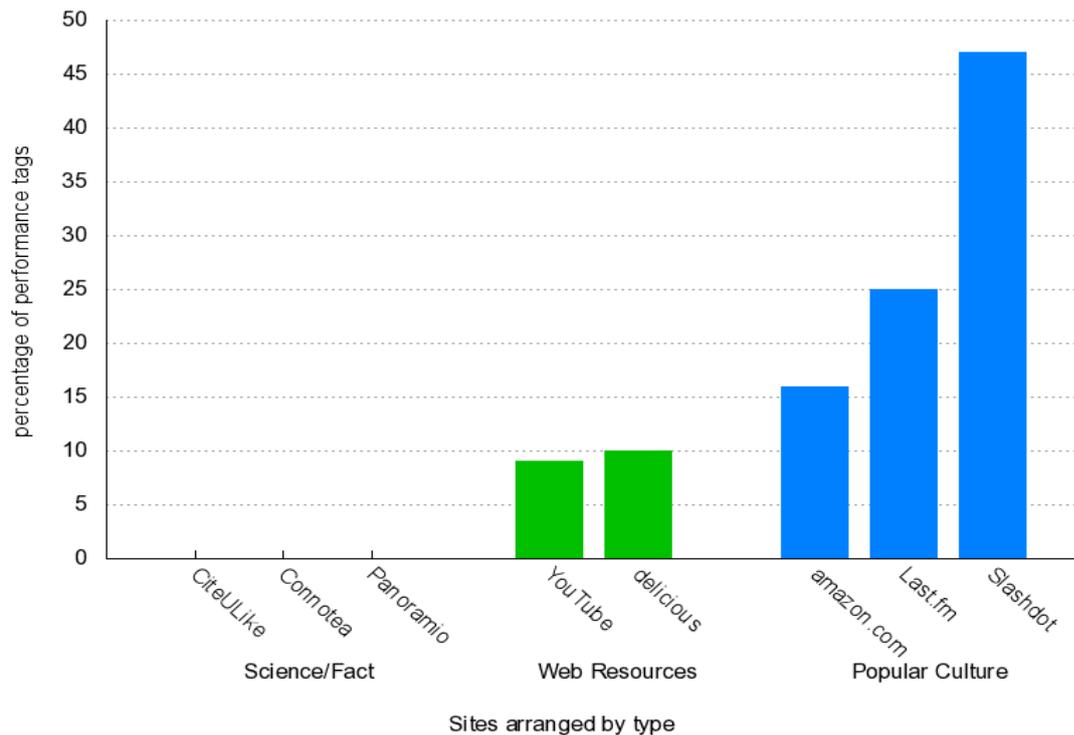


Figure 1: The variation in percentage of performance tags, compared to the theme of each web site.

Given these results, we make the following conclusions: the internal structure of performance tags differs greatly inter-and intra- tagging sites and tagger groups, and performance tags are not easily identified by structure alone. This preliminary study having shown that there is a wide range of visible variation, we expect to widen the study further as part of our future work. However, we found during this study that a prerequisite to automated use of performance tags, and indeed to the analysis of tags in general, is the manual post-coordination of terms. Hence, an automated approach to post-coordination of terms is considered to be a useful tool in analysis or practical use of tagsets. We will therefore now examine the problem of post-coordination of tags.

Study 2: Post-coordination of tags

With the manual approach described above as a model, we can describe the process of manually post-coordinating tags. This process could be described as a form of segmentation – either sentence segmentation, in the event that the tag set is made up of full sentences, or phrase segmentation.

This bears some similarity to the problem of segmenting compound terms in tags in which the intra-word boundaries have been lost (Tonkin, 2006). The desired behaviour in this case is the detection of 'phrase boundaries', rendered more complex in this case by the fact that full annotations are often only sentence fragments. A very similar problem is referred to in the area of information retrieval as 'query segmentation' (Bergsma & Wang, 2007). Query segmentation is defined as the process of taking a user's search-engine query and dividing the tokens into individual phrases or semantic units. Bergsma and Wang suggest that this may improve precision in searches, since an understanding of the appropriate segmentation can allow ranking of query results that privilege those on which the correct form of the phrase occurs. They also suggest that recall may be improved, since queries may be expanded or substituted by semantically similar alternatives.

Consider for example the tagset present in Fig. 2. Each of these tagsets contains a number of phrases; the second, for example, contains 'baby shower' and 'you tube'. The third contains 'video game' and 'performing arts'. The last can be read as three phrases; (five brothers) (genetic engineering) (weird rich people).

york new mahattan brooklyn coney island empire state chrysler building central park flatiron
statue

girl smile funny warcraft random (baby shower) music (you tube)
(film advertising) tv (video game) commercials web series entertainment news (performing arts)
(five brothers) (genetic engineering) (weird rich people)

Figure 2: Tag sets taken from YouTube

As is frequently the case with analysis after the fact, it is not possible in every case to identify the 'true' segmentation or post-coordination of these tags; to demonstrate this, consider the following example, for which there are two valid parses:

september 2nd wife

Therefore, it is not possible to define a rule that optimally covers all possibilities. We must therefore limit ourselves to examining plausible options – that is, to find the most probable segmentations. There are various approaches extant to query segmentation – Bergsma and Wang, for example, describe an approach based around a support vector machine for classification. For our purposes, we demonstrate a very simple approach, using Markov chains learnt from an existing data corpus. This approach bears certain similarities to that described by Risvik et al (2003).

Markov chains and probability

Markov chains are perhaps best known for autogenerating plausible-sounding nonsense text, to the extent that the assumption is often made that their use should be limited to this purpose. In reality, they may be used to describe any processes that are governed by probability, but in which each subsequent step depends solely on the current state of the system. Language is not really one of these processes, since in fact the validity of inserting each subsequent word in a sentence generally depends on all the other terms in the sentence, and possibly on other variables as well. However, this assumption provides a very simple and cheap early approximation, so is not an unreasonable first step.

The general description of Markov chains may be given as follows; consider a system in which there are S states, $S = \{s1, s2...sn\}$. Our process begins in a given state, and then moves from that state to another, once each timestep. There is a probability associated to each transition – that is, some changes are more likely than others. Howard (1971) illustrates this via the classic example of a frog on a lily pad, which begins on one lily pad and then hops to another, and then another. The frog is quite likely to move to another lily pad that is within easy hopping distance. However, it is very unlikely to leap from one end of the pond to the other in a single step. Our use of Markov chains will make use of this fact to determine the most likely course of events – where the disconnects occur. With our metaphorical frog, a very unlikely transition might imply that it swam or was carried to another lily pad, thus breaking up the pattern of jumps and causing a break in our observations. It should be possible to guess at where these breaks occur.

In our case, the states – the 'lily pads' – are the words in the tag set. The 'distance' between words represents the likelihood that they occur together. For example, the words 'New York' are often seen together. On the other hand, the words 'motherboard smile' have a low probability of being directly linked, although they may be associated by a less direct relationship and may therefore appear in the same tag set.

For our purpose, we need information about the probability of transition between the words in the tag set, and information about the probability that a split will occur. When we perform this task manually, we use our own stored knowledge about language to make the judgment. An automated system, on the other hand, needs to collect this information in some manner. The possibilities for collecting tailored transition properties will be discussed later.

Building a ground truth

Because in this instance no 'correct' solution can be derived, a gold standard (that is, a set of manually split entries which are considered to be optimally split) is required for evaluation of our approach. This simply requires a set of tags to be taken and appropriately annotated. Potentially, a ground truth is also useful as a training set, since it allows the estimation of variables such as: the number of tags that are part of a larger structure, that is, the actual frequency of occurrence of tags requiring postcoordination; and the type of structures that are seen. For example, our examples (Fig. 2) suggest that the frequency of multi-word terms is quite high, with terms such as 'New York' and 'video game', but only one potential performance tag ('weird rich people') is

visible, and we have seen no examples of a grammatical sentence. In fact, these do appear in YouTube tagsets – two randomly chosen sample constructions being 'so this is love' and 'hip hop saved my life' – although they are relatively infrequent.

For our purposes, the gold standard was manually created by one individual, and therefore represents only a guess. This is potentially a serious source of inaccuracy. Post-coordination of tags without recourse to examining the tagged resource is in itself a very difficult problem. Correct segmentation, especially where any sort of specialist terminology is used – which in popular culture is very often the case – may depend on common-sense or specialised background knowledge.

There exist a number of possible approaches to segmentation on the basis of Markov chains. The obvious approach, and the one that we will examine here, is the brute-force approach in which we simply make use of a large corpus of known documents (in this case, we will use the Simple English Wikipedia as a basis for this work, simply because it is a relatively small and immediately available data source). In this approach, we learn transitions in the following manner:

```
For each word in input stream:
  If Current_Word has not been seen before, create an entry for it.
  Create 1 link between Previous_Word and Current_Word
  If Current_Word has been seen before:
    Increment number of links between Previous_Word and Current_Word
```

Figure 3: Basic pseudocode for constructing a Markov chain
(excluding normalisation step)

Sentence boundaries, brackets, clauses and similar transitions can create problems; therefore, one might choose to create a 'stop' symbol to represent end of sentence, rather than creating spurious entries when one sentence ends and another begins (ie. 'He saw a fly. Magazines such as ...' might otherwise result in an entry linking 'fly' and 'magazines' as a phrase). If the likely location of sentence breaks is not useful, however, there is no need to register it; rather, one may simply throw away that information.

This approach has familiar drawbacks; specifically, the choice of corpus from which to learn transition probabilities and term frequencies dramatically alters the effectiveness of the approach. One potential solution to this is the use of specialised corpora – for example, to make use of tagsets, one might look at the links that exist between the indicated resource and other, related resources. If some of these resources are texts, then one might consider using these as a basis for analysis. An alternative approach might be to make use of indirect analysis using a set of knowledge bases: for example, one might consider identifying types (firstname, lastname) from a dataset; well known phrases might be identified by making use of wordnet or a similar type of knowledge base; even parts of speech (adjectives, nouns, etc) could potentially be useful in annotating known terms.

Possible combinations of tags

A given set of tags might be segmented in any of a number of combinations: for example, there are eight combinations of four tags (see Figure 4).

a b c d
 (a b) c d
 a (b c) d
 a b (c d)
 (a b) (c d)
 (a b c) d
 a (b c d)
 (a b c d)

Possible Combinations

1
 1 1
 1 2 1
 1 3 3 1
 1 4 6 4 1
 1 5 10 10 5 1

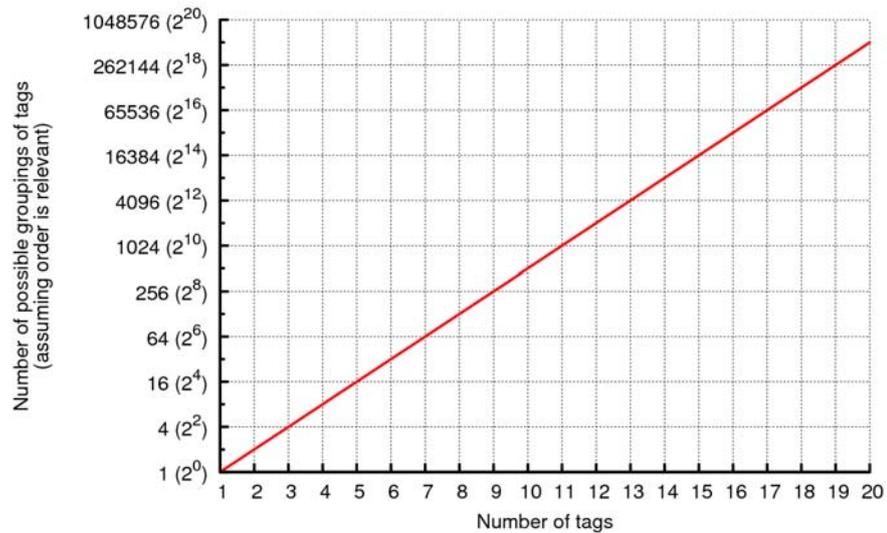


Figure 4: Combinations of tags

The number of combinations is predictable using Pascal's triangle; in this case, the fourth line (1+3+3+1=8). It is evident from looking at further rows that the number of combinations grows quickly, being governed by the following equation; if m is the number of terms per line of tags, n is $m-1$ (see graph, Fig. 4):

$$\sum_{k=0}^n \binom{n}{k} = \sum_{k=0}^n \frac{n!}{k!(n-k)!} = 2^n$$

Therefore it is desirable, particularly with larger sets of terms, to avoid a direct brute-force examination of all combinations. There exist various strategies for avoiding this; however, here we have largely accepted a brute-force approach.

The combinations given (Fig. 4, left) can be generated in two logical stages:

- 1) work out all possible combinations of integers that can be added together to total n .
- 2) describe all possible unique ordered permutations of these integers (ie. 1 1 2, 1 2 1, 2 1 1 are unique permutations of the integers 1, 1 and 2).

Figure 5: Pseudo-code for finding the partitions of the integers (ZS2)

The first of these component tasks is known as *finding the partitions of the integers*. This can be solved using an existing implementation, such as Perl's Integer::Partition, an implementation of the Zoghbi and Stojmenovic (1998) ZS1 and ZS2

algorithms for generating integer partitions. The pseudocode of the ZS2 algorithm is replicated in Figure 5, above.

The second subtask is solved using a list permutation algorithm to get the different orders in which these terms may appear (such as (2,1) and (1,2)), again, all duplicates, if generated by the implementation used, get removed (permutation algorithms may give several instances of each combination). This is a fairly brute force and inefficient way of achieving our goal, but for the purpose of a proof of concept, it is a very quick implementation.

Note that, as mentioned above, the number of permutations rises rapidly (exponentially). In practice, the number of possible permutations to be generated can be limited relatively cheaply by pre-processing the input string, seeking and removing tags that are not well-formed (ie are not recognisable words or symbols), on the principle that as we do not recognise them, we already know that we have no knowledge about how these tags might combine with others. This has the benefit of producing smaller and more manageable fragments, and massively reducing the number of combinations.

Scoring combinations

Once all the possible combinations of terms are known, then it is possible to evaluate the probability that each combination may arise, according to the n-grams that we have learnt. Consider the following example:

(five brothers) (genetic engineering) (weird rich people)

five + brothers appears 3 times
brothers + genetic appears 0 times
genetic + engineering appears 13 times
engineering + weird appears 0 times
weird + rich appears 0 times
rich + people appears 39 times

Ignoring the normalisation step, simply using these raw numbers as information sources to abolish links between words gives a creditable partial parse:

(five brothers) (genetic engineering) weird (rich people) Combination 1

However, in practice there are several possible configurations, some of which score more highly than others. Our approach must weigh up the likelihood that terms are linked (how frequently 'less than atomic' terms, ie, phrases, actually occur in this dataset) against the statistical information available about linking between specific examples of words.

Using the list of combinations collected above, it is possible to calculate the probability of each one. Using the above example, it is clear that

(five) (brothers genetic) (engineering weird rich) (people) Combination 2

is a very improbable combination. Without involving probabilities, this may be calculated in the following way; setting a threshold value V_t to represent the likelihood that a split occurs, such as 2 (to ensure that very uncommon transitions are not represented), we may add raw link counts for each combination:

$$\text{Combination 1} = 3 + V_t + 13 + V_t + V_t + 39 = 61$$

$$\text{Combination 2} = V_t + 0 + V_t + 0 + 0 + V_t = 6$$

In Markov terms, one might calculate the n -step transition probability; however, one must bear in mind that as partial matches are required, it is necessary to reflect the probability that a split occurs in calculating the probability of each transition. Normalisation into probabilities is not automatically useful in this scenario; very frequently and widely used terms (for example, words such as 'new') attain a low probability for any given combination as there are so many unique possibilities. The threshold value may be set in a manner proportional to the number of possibilities in order to offset this effect. Ideally, it should represent the actual probability that a split occurs after each term; however, this value requires a substantial training set to calculate effectively. In future work this could be calculated using a large set of tags from a site from which well-formed multi-word phrasal tags

are available; training from a different dataset may lead to inaccuracies, however, as the use of language may differ greatly between sites.

Evaluation

The basic approach that we describe above, making use of the Simple English Wikipedia as a learning tool, is evaluated against the ground truth according to the following rules: we examine the five most highly scored options (ie. most probable combinations of terms). If the correct solution is present as one of the top five, then we describe it as 'correct', but weight that value according to the position in the top five. If it is not present in its entirety, but one of the features noted has been correctly identified, then it is described as 'partially correct', and again is weighted according to the position of that result in the top five list. If no part of the solution is present, then the result is considered to be incorrect.

Evaluating against a hundred tag 'sentences', we found that approximately one in ten of them received an entirely incorrect parse. A further 23% received a partially correct parse. Figure 6 shows the variation in parsing accuracy against the number of terms in the overall expression, the mean accuracy and a linear regression that demonstrates reduction in accuracy as the number of terms rises.

The Receiver Operating Characteristics curve shown on the right of this figure demonstrates the uneasy relationship between sensitivity and accuracy – as the sensitivity of the algorithm is increased, so does the number of false positives. As the sensitivity decreases, a larger number of false negatives results. This also demonstrates that the test is not particularly effective – at best, the performance of this simple approach could be described as fair. The limited precision of our ground truth, however, may have some impact on this. In future, one might instead use a scoring system that depended instead on evaluating how 'reasonable' each grouping seems to the user.

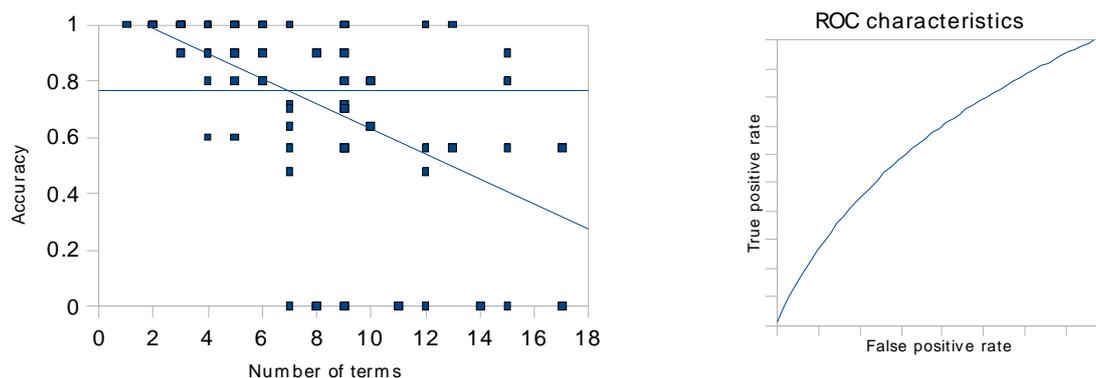


Figure 6: Left: a sample showing the variation of parsing accuracy with increase in numbers of terms (low threshold for combination, maximising sensitivity over accuracy).

Right: ROC characteristics for the system as a classifier, applying a number of different threshold weights.

There exist a wide range of linguistic constructions (and words) visible on the many tagging services tested. Hence, a full evaluation was considered to be beyond the scope of this paper, and will comprise part of our future work in this area. In particular, it is notable that the choice of n-gram corpus on which to train will have a severe impact on the accuracy of a given system. It was noticeable that this particular parser tended to fail on current affairs issues and on sales terminology, perhaps because the Simple English Wikipedia simply does not contain a great deal of either.

One might make use of a very large data source such as Google's n-gram corpus, along with other data sources such as cleaned lists of locations and names, all appropriately weighted to take into account the origin of the information. Additionally, linguistic information such as the grammatical plausibility of the construct – which this approach simply does not take into account – would potentially be of use, in particular in the case in which individuals use coherent structures, such as noun-phrase constructions. The effectiveness of this approach might again be expected to vary according to the site.

Discussion

Having demonstrated that tags can potentially be both pre-and post-coordinated via machine learning methods, although the results may be limited in accuracy for various reasons, the question becomes discovery of potential applications for this normalisation process. Here we have discussed the possibility that identification and development of appropriate 'readings' for performance tags might constitute one such application.

Our initial study suggested that simple heuristics may be somewhat successful in identifying performance tags; for example, the length of the tag can in itself be an indicator. We find, however, that this heuristic is not very accurate; on Slashdot, for example, the distinction in average tag length was approximately 0.6, which is barely statistically significant given the standard deviation.

There are many other candidate approaches that may be useful in identification of performance. For example, one approach might be to examine the structure of performance tags by comparison to other tags (eg. grammatical structure). Another might be to capitalise on the fact that performance tags often contain an evident element of affect (are designed to be emotive), and are therefore potentially susceptible to sentiment analysis. Sentiment analysis or opinion mining (Pang & Lee, 2008) as a field examines the question of determining the writer's/tagger's feelings on an issue, and therefore it could be seen as an inversion of the intended use of these tools to apply them in an attempt to find out whether the writer intended to place any emotional payload in the tag.

It is perhaps reasonable to expect that knowledge-based approaches will be much less successful in identifying performance tags than in many other areas. By their very nature, these tags are intended to make a point about the author in a dramatic or emotionally involving manner, and therefore performance tags often play on the reader's emotions or appeal to their understanding of joint cultural references. It is difficult to identify humour, just as it is difficult to identify sarcasm or self-referential speech, without the ability to 'mind-read', to place oneself in the position of the author and to imagine what they might have meant. The ability to guess at the author's intended reading may depend on a good knowledge of the topic area, and hence is likely to be quite a challenge.

The variation in appearance of performance tags, and indeed of tag syntax in general, raises further questions about the factors influencing tag choice and structure. Sites that publicly display tags in proximity to content such as product or artist descriptions tend to receive more performance tags, suggesting a socially-motivated pattern of use. The existence of tag sharing/browsing facilities alone does not have this effect.

Conclusion

We have seen that it is possible to coordinate tags using a number of machine learning methods taken from other domains, such as the area of query segmentation. One area for future work in this domain is the development of better and more accurate approaches toward this task, based perhaps on linguistic category classification, alternative machine learning approaches, and better training. Another area is that of applying this work to real-world interfaces, both during tag production and during reading and search; for example, on typing in a set of tags an application may visually show how they are coordinated in such a way that the user may correct that judgement.

Performance tags as a topic have other interesting side effects, firstly in that they bring the problem of judging affect into tagging, and secondly in that they bring to the fore the question of the interactive process of reading, an ongoing theme in the area of tagging research. Finally, our analysis has discussed the very wide variation on a number of axes present across various tagging sites, which suggests the significant point that, in making judgements about social tagging based on a single sample or set of samples, we err as significantly as when taking samples of language use from a single location (such as a supermarket till) as indicative of the characteristics of that language as a whole.

Bibliography

Bergsma, S. and Wang, Q. I. (2007). Learning Noun Phrase Query Segmentation, In EMNLP-CoNLL 2007, Prague, Czech Republic, June 28-30, 2007, pages 819-826.

Catarino, M. E. and Baptista, A. A. (2008) Social tagging and Dublin core: a preliminary proposal for an application profile for DC social tagging, ELPUB2008. Open Scholarship: Authority, Community, and Sustainability in the Age of Web 2.0 – Proc. 12th International Conference on Electronic Publishing, Toronto,

Canada 25-27 June 2008 Ed.: Leslie Chan and Susanne Mornatti. ISBN 978-0-7727-6315-0, 2008, pp. 100-110

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

Gray, J. (2003). New Audiences, New Textualities: Anti-Fans and Non-Fans. *International Journal of Cultural Studies*, 6(64). DOI: 10.1177/1367877903006001004

Kipp, M. I. (2007). @toread and Cool: Tagging for Time, Task and Emotion. Proceedings 8th Information Architecture Summit. Las Vegas, Nevada, USA: Retrieved September 30, 2008 from Web site: <http://eprints.rclis.org/archive/00010445/>.

Landgren, D. (2007). Integer::Partition – Generate all integer partitions of an integer. Version 0.03. Retrieved September 30, 2008 from <http://search.cpan.org/~dland/Integer-Partition-0.03/Partition.pm>

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2):1—135.

Phoenix, T. (1999). List::Permutor - Process all possible permutations of a list. Version 0.022. Retrieved September 30, 2008 from <http://search.cpan.org/~phoenix/List-Permutor-0.022/>

Risvik, K. M., Mikolajewski, T. and Boros, P. (2003). Query segmentation for web search. In WWW (Poster Session).

Schechner, R. (2001). *Performance studies: An Introduction*. New York: Routledge.

Tonkin, E. (2006). Searching the long tail: Hidden structure in social tagging. In Furner, Jonathan and Tennis, Joseph T., Eds. Proceedings 17th Workshop of the American Society for Information Science and Technology Special Interest Group in Classification Research 17, Austin, Texas.

Tonkin, E. (2008). Orthography, Structure and lexical choice as identity markers in social tagging environments. IADIS International Conference on Web Based Communities

Yeung, C. A., Gibbins, N. and Shadbolt, N. (2007). Mutual Contextualization in Tripartite Graphs of Folksonomies. ISWC/ASWC 2007: 966-970

Zoghbi, A. and Stojmenovic, I. (1998). Fast algorithms for generating integer partitions. *Intern. J. Computer Math.*, Vol- 70. pp. 319 332.

Zöllers, A. (2007). Emerging Motivations for tagging. Retrieved from Expression, Performance, and Activism: http://www2007.org/workshops/paper_55.pdf.