Context and Domain Knowledge Enhanced Entity Spotting In Informal Text

Daniel Gruhl[1], Meena Nagarajan[2], Jan Pieper[1], Christine Robson[1], Amit Sheth[2]

[1] IBM Almaden Research Center
650 Harry Road, San Jose, CA
{dgruhl, jhpieper, crobson}@us.ibm.com

[2] Knoesis, 377 Joshi Research Center
3640 Colonel Glenn Highway, Dayton, OH
{meena, amit}@knoesis.org

Abstract. This paper explores the application of restricted relationship graphs (RDF) and statistical NLP techniques to improve named entity annotation in challenging Informal English domains. We validate our approach using on-line forums discussing popular music. Named entity annotation is particularly difficult in this domain because it is characterized by a large number of ambiguous entities, such as the Madonna album "Music" or Lilly Allen's pop hit "Smile".
We evaluate improvements in annotation accuracy that can be obtained by restricting the set of possible entities using real-world constraints. We find that constrained domain entity extraction raises the annotation accuracy significantly, making an infeasible task practical. We then show that we can further improve annotation accuracy by over 50% by applying SVM based NLP systems trained on word-usages in this domain.

1 Introduction

The semantic web and the plethora of relationships expressed as RDF files provide a wealth of information as to how entities in a document might relate. However, in the absence of a training corpus with in-line references to the entities (a "pre-annotated corpus"), it becomes difficult to identify and disambiguate named entities in text[13] to leverage these relationships in more complex tasks. The mapping of regions of text to entries in an ontology becomes harder when the regions are words used commonly in everyday language, such as "Yesterday," which could refer to the previous day, a Beatles song (one of 897 songs with that title), or a movie (there are three productions so named).

Sense disambiguation (the process of identifying which meaning of a word is used in any given context) becomes even more challenging when there is insufficient context surrounding the discourse; the language used is in the Informal English domain common to social networking sites – a blend of abbreviations, slang and context dependent terms delivered with an indifferent approach to

grammar and spelling. Understanding the semantic relationships between entities in these challenging domains is necessary for a variety of information-centric applications, including the BBC SoundIndex [1]. This application, developed by the authors and others, provides a realtime "top 40" chart of music popularity based on sources such as MySpace and YouTube.

If we wish to utilize this type of content we need to transform it into a structured form by identifying and sense disambiguating particular entities such as mentions of artists, albums and tracks within the posts. In this paper we explore how the application of domain models (represented as a relationship graph, e.g., RDF) can complement traditional statistical NLP techniques to increase entity spotting[3] accuracy in informal content from the music domain. Semantic annotation of track and album name mentions are performed with respect to MusicBrainz RDF[4] - a knowledge base of instances, metadata and relationships in the music domain. An example snapshot of the MusicBrainz RDF is shown in Figure 1.

1.1 Challenging features of the Music Domain

Availability of domain models is increasingly common with today's many Semantic Web initiatives. However, employing them for annotating Informal English content is non-trivial, more so in the music domain (see Table 1). Song titles are often short and ambiguous. Songs such as "The" (four songs), "A" (74 songs), "If" (413 songs), and "Why" (794 songs) give some idea of the challenges in spotting these entities. In annotating occurrence of these elements in text, for example, 'Yesterday' in "loved your song Yesterday!", we need to identify which entity 'Yesterday', among the many in the ontology, this one refers to.

Bands with a song "Merry Christmas"	60
Songs with "Yesterday" in the title	3,600
Releases of "American Pie"	195
Artists covering "American Pie"	31

Table 1. Challenging features of the music domain.

Here, we present an approach that systematically expands and constrains the scope of domain knowledge from MusicBrainz used by the entity spotter to accurately annotate such challenging entity mentions in text from user comments. The MusicBrainz data set contains 281,890 artists who have published at least one track and 4,503,559 distinct artist/track pairs.

1.2 Our Approach and Contributions

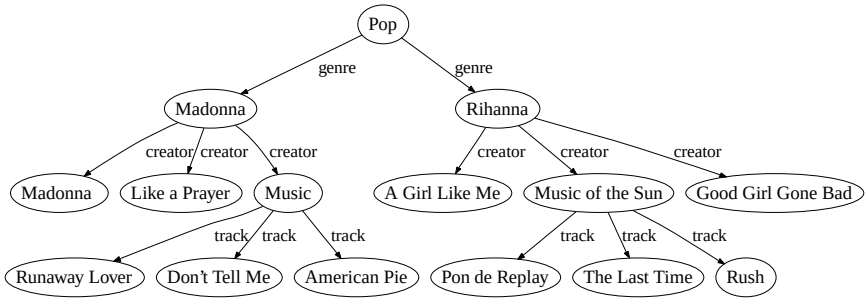
We begin with a light weight, edit distance based entity spotter that works off a constrained set of potential entities from MusicBrainz. The entities we are interested in spotting in this work are track, album and song mentions. We constrain the size of the set of potential entities by manually examining some of the restrictions that can be applied on the MusicBrainz ontology. Restrictions

[3] We define spotting as finding a known list of named entities in text in real-time.
[4] http://wiki.musicbrainz.org/RDF

are obtained using additional information from the context of the entity spot. For example, when considering a spot in a comment from a discussion group on country music, we may only consider artists and songs from that genre.

Further improvement is needed to disambiguate the usage of song titles. For example, while Lilly Allen has a song titled 'Smile,' not all mentions of this word on her MySpace page refer to the song, for example, "your face lights up when you smile". We disambiguate the output of our naive spotter with more advanced NLP techniques using an SVM classifier that takes into account the characteristics of word usages.

We find that constraining the domain of possible entity matches before spotting can improve precision by several orders of magnitude over an admittedly poor baseline of the light weight spotter. We note that these improvements follow a Zipf distribution, where a reduction of possible entity matches by 50% equals a doubling of precision. We also find that use of our NLP system can improve accuracy by more than another 50%. These two steps, presented in the rest of this paper, can form the beginning of a processing pipeline to allow higher precision spot candidates to flow to upstream applications.




```
I went to <artist id=89>Madge's</artist> concert last night.
<artist id=262731>Rihanna</artist> is the greatest!
I love <artist id=357688>Lily's</artist> song <track id=8513722>smile</track>.
```

Fig. 1. RDF Snapshot of MusicBrainz and example of in-line annotations. These annotations illustrate how messages in our corpus can be tagged with universally unique identifiers (in this case the MusicBrainz id number) to facilitate searches both for individual mentions as well as Business Intelligence style roll-ups of aggregate statistics on mentions in the corpus.

2 Related Work

2.1 Named Entity Recognition and use of Domain Knowledge

Named Entity Recognition (NER) is an important task in information extraction. Nadeau and Sekine present a comprehensive survey of NER since 1991 [15]. The KnowItAll Information Extraction system [8] makes use of entity recognition techniques, in a domain-independent fashion. Related work by Chieu and Ng has shown high performance in entity extraction with a single classifier and information from the whole document to classify each word [6].

Closely related to our work, domain dictionaries have been widely used in NER, including Wikipedia[4] and Wiktionary [14], DBLP [10], KAON [3], and MusicBrainz [1]. They have also been used for the task of disambiguating entity senses, an important step in accurately extracting entities. Work in [4] exploited the link and textual features of Wikipedia to perform named entity disambiguation. Entity disambiguation by gathering context from the document and comparing it with context in the knowledge base was also explored in [10].

These provide inspiration for our work, demonstrating that it is possible to do efficient and accurate NER on a document-by-document basis using domain knowledge supplemented with natural language processing techniques. Our work differs in how we constrain a domain knowledge base in order to annotate a set of known named entities in Informal English content.

2.2 Named Entity Recognition in Informal English

The challenge of NER in noisy and informal text corpora has been explored from several angles. Minkov et al. were the first to address NER in "informal text" such as bulletin board and newsgroup postings, and email [13]. Their work on recognizing personal names in such corpora is particularly relevant, as it uses dictionaries and constraining dictionary entries. They use a TF/IDF based approach for constraining the domain space, an approach we considered in early versions of our music miner. However, we found this approach to be problematic in the music domain, as song titles often have very low novelty in the TF/IDF sense (e.g. the Beatles song, "Yesterday"). Work by Ananthanarayanan et al. has also shown how existing domain knowledge can be encoded as rules to identify synonyms and improve NER in noisy text [2].

Our approach to NER in informal text differs in that it is a two step process. Given a set of known named entities from the MusicBrainz RDF, we first eliminate extraneous possibilities by constraining the domain model using available metadata and further use the natural language context of entity word-usages to disambiguate entities that appear as entities of interest and those that do not. Some word-usage features we employ are similar to those used in the past [13], while others are derived from our domain of discourse.

3 Restricted Entity Extraction

We begin our exploration of restricted RDF graphs or Ontologies to improve entity spotting by investigating the relationship between the number of entities (artists, songs and albums) considered for spotting and the precision of the entity spotter. The result is a calibration curve that shows the increase in precision as the entity set is constrained. This can be used to gauge the benefit of implementing particular real world constraints in annotator systems. For example, if detecting that a post is about an artist's recent album requires three weeks of work, but only provides a minor increase in precision, it might be deferred in favor of an "artist gender detector" that is expected to provide greater restriction in most cases.

3.1 Ground Truth Data Set

Our experimental evaluation focuses on user comments from the MySpace pages of three artists: Madonna, Rihanna and Lily Allen (see Table 2). The artists were selected to be popular enough to draw comment but different enough to provide variety. The entity definitions were taken from the MusicBrainz RDF (see Figure 1), which also includes some but not all common aliases and misspellings.

Madonna	an artist with a extensive discography as well as a current album and concert tour
Rihanna	a pop singer with recent accolades including a Grammy Award and a very active MySpace presence
Lilly Allen	an independent artist with song titles that include "Smile," "Allright, Still", "Naive", and "Friday Night" who also generates a fair amount of buzz around her personal life not related to music

Table 2. Artists in the Ground Truth Data Set

We establish a ground truth data set of 1858 entity spots for these artists (breakdown in Table 3). The data was obtained by crawling the artist's MySpace page comments and identifying all exact string matches of the artist's song titles. Only comments with at least one spot were retained. These spots were then hand scored by four of the authors as "good spot," "bad spot," or "inconclusive." This dataset is available for download from the Knoesis Center website [5].

Artist (Spots scored)	Good spots Agreement		Bad spots Agreement	
	100%	75 %	100%	75%
Rihanna (615)	165	18	351	8
Lily (523)	268	42	10	100
Madonna (720)	138	24	503	20

Table 3. Manual scoring agreements on naive entity spotter results.

The human taggers were instructed to tag a spot as "good" if it clearly was a reference to a song and not a spurious use of the phrase. An agreement between at least three of the hand-spotters with no disagreement was considered agreement. As can be seen in Table 3, the taggers agreed 4-way (100% agreement) on Rihanna (84%) and Madonna (90%) spots. However ambiguities in Lily Allen songs (most notably the song "Smile"), resulted in only 53% 4-way agreement.

We note that this approach results in a recall of 1.0, because we use the naive spotter, restricted to the individual artist, to generate the ground truth candidate set. The precision of the naive spotter after hand-scoring these 1858 spots was 73%, 33% and 23% for Lilly Allen, Rihanna and Madonna respectively (see Table 3). This represents the best case for the naive spotter and accuracy drops quickly as the entity candidate set becomes less restricted. In the next Section we take a closer look at the relationship between entity candidate set size and spotting accuracy.

[5] http://knoesis.wright.edu/research/semweb/music

3.2 Impact of Domain Restrictions

One of the main contributions of this paper is the insight that it is often possible to restrict the set of entity candidates, and that such a restriction increases spotting precision. In this Section we explore the effect of domain restrictions on spotting precision by considering random entity subsets.

We begin with the whole MusicBrainz RDF of 281,890 publishing artists and 6,220,519 tracks, which would be appropriate if we had no information about which artists may be contained in the corpus. We then select random subsets of artists that are factors of 10 smaller (10%, 1%, etc). These subsets always contain our three actual artists (Madonna, Rihanna and Lily Allen), because we are interested in simulating restrictions that remove invalid artists. The most restricted entity set contains just the songs of one artist (≈0.0001% of the MusicBrainz taxonomy). In order to rule out selection bias, we perform 200 random draws of sets of artists for each set size - a total of 1200 experiments. Figure 2 shows that the precision increases as the set of possible entities shrinks. For each set size, all 200 results are plotted and a best fit line has been added to indicate the average precision. Note that the figure is in log-log scale.

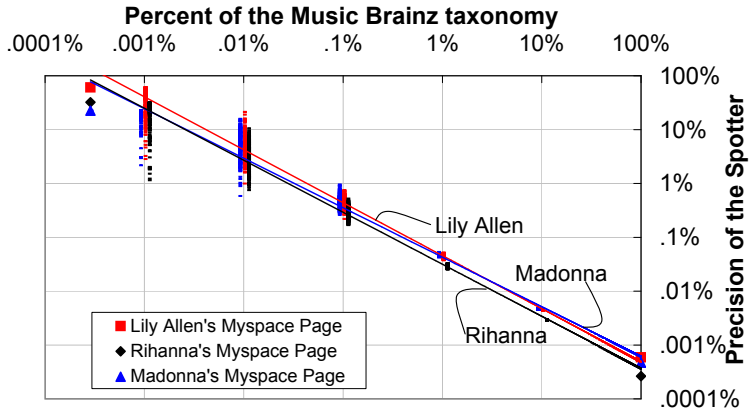


Fig. 2. Precision of a naive spotter using differently sized portions of the MusicBrainz Taxonomy to spot song titles on artist's MySpace pages

We observe that the curves in Figure 2 conform to a power law formula, specifically a Zipf distribution ($\frac{1}{n^{R^2}}$). Zipf's law was originally applied to demonstrate the Zipf distribution in frequency of words in natural language corpora [18], and has since been demonstrated in other corpora including web searches [7]. Figure 2 shows that song titles in Informal English exhibit the same frequency characteristics as plain English. Furthermore, we can see that in the average case, a domain restrictions of 10% of the MusicBrainz RDF will result approximately in a 9.8 times improvement in precision of a naive spotter.

This result is remarkably consistent across all three artists. The R^2 values for the power lines on the three artists are 0.9776, 0.979, 0.9836, which gives a deviation of 0.61% in R^2 value between spots on the three MySpace pages.

4 Real World Constraints

The calibration results from the previous Section show the importance of "ruling out" as many artists as possible. We observe that simple restrictions such as gender that might rule out half the corpus could potentially increase precision by a factor of two. One way to impose these restrictions is to look for real world constraints that can be identified using the metadata about entities as they appear in a particular post. Examples of such real world constraints could be that an artist has released only one album, or has a career spanning more than two decades.

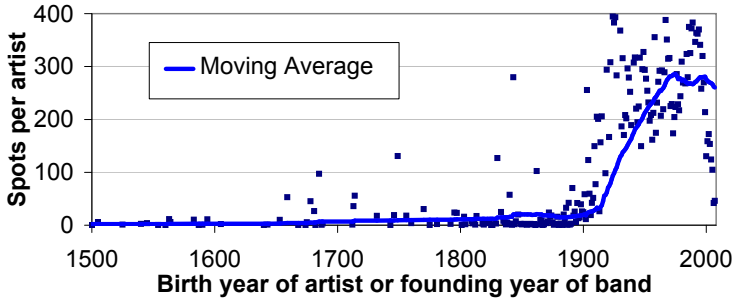


Fig. 3. Songs from all artists in our MySpace corpus, normalized to artists per year.

We are interested in two questions. First, do real world constraints reduce the size of the entity spot set in a meaningful way? Second, by how much does the trivial spotter improve with these real world constraints and does this match with our predicted improvements from Figure 2? The effect of restricting the RDF by artist's age can be seen in Figure 3, which shows spots per artist by birth date. Interestingly, we can see a spike in the graph beginning around 1920 with the emergence of Jazz and then Rock and Roll, reflecting the use of common words as song titles, (e.g. "Blues" and "South" by Louis Armstrong). For all artists since this date (94% of the MusicBrainz Ontology, and 95.5% of the naive spots on our corpus), the increased use of natural language utterances as song titles is evidence that we should expect the Zipf distribution to apply to any domain restriction over the corpus.

Having established that domain restrictions do reduce spot size, we look for further constraints that can be inferred from the user-generated text. As an example, we observe that comments such as "Saw you last night in Denver!!!" indicate the artist is still alive. A more informational post such as "Happy 25th B-DAY!" would allow us to further narrow the RDF graph to 0.081% of artists in the Ontology, and 0.221% of the naive spots on Lily Allen's MySpace Page.

Our constraints are tabulated in Table 4, and are derived manually from comments such as, "I've been a fan for 25 years now," "send me updates about your new album," and "release your new album already! i'm getting tired of playing your first one on repeat!" Since we have chosen our corpus to represent three specific artists, the name of the artist is a further narrowing constraint.

Key	Count	Restriction
Artist Career Length Restrictions- Applied to Madonna		
B	22	80's artists with recent (within 1 year) album
C	154	First album 1983
D	1,193	20-30 year career
Recent Album Restrictions- Applied to Madonna		
E	6,491	Artists who released an album in the past year
F	10,501	Artists who released an album in the past 5 years
Artist Age Restrictions- Applied to Lily Allen		
H	112	Artist born 1985, album in past 2 years
J	284	Artists born in 1985 (or bands founded in 1985)
L	4,780	Artists or bands under 25 with album in past 2 years
M	10,187	Artists or bands under 25 years old
Number of Album Restrictions- Applied to Lily Allen		
K	1,530	Only one album, released in the past 2 years
N	19,809	Artists with only one album
Recent Album Restrictions- Applied to Rihanna		
Q	83	3 albums exactly, first album last year
R	196	3+ albums, first album last year
S	1,398	First album last year
T	2,653	Artists with 3+ albums, one in the past year
U	6,491	Artists who released an album in the past year
Specific Artist Restrictions- Applied to each Artist		
A	1	Madonna only
G	1	Lily Allen only
P	1	Rihanna only
Z	281,890	All artists in MusicBrainz

Table 4. The efficacy of various sample restrictions.

We consider three classes of restrictions - career, age and album based restrictions, apply these to the MusicBrainz RDF to reduce the size of the entity spot set in a meaningful way and finally run the trivial spotter. For the sake of clarity, we apply different classes of constraints to different artists.

We begin with restrictions based on length of career, using Madonna's MySpace page as our corpus. We can restrict the RDF graph based on total length of career, date of earliest album (for Madonna this is 1983, which falls in the early 80's), and recent albums (within the past year or 5 years). All of these restrictions are plotted in Figure 4, along with the Zipf distribution for Madonna from Figure 2. We can see clearly that restricting the RDF graph based on career characteristics conforms to the predicted Zipf distribution.

For our next experiment we consider restrictions based on age of artist, using Lily Allen's MySpace page as our corpus. Our restrictions include Lily Allen's age of 25 years, but overlap with bands founded 25 years ago because of how dates are recorded in the MusicBrainz Ontology. We can further restrict using album information, noting that Lily Allen has only a single album, released in the past two years. These restrictions are plotted in Figure 4, showing that these restrictions on the RDF graph conform to the same Zipf distribution.

Finally, we consider restrictions based on absolute number of albums, using Rihanna's MySpace page as our corpus. We restrict to artists with three albums, or at least three albums, and can further refine by the release dates of these albums. These restrictions fit with Rihanna's short career and disproportionately large number of album releases (3 releases in one year). As can be seen in Figure 4, these restrictions also conform to the predicted Zipf distribution.

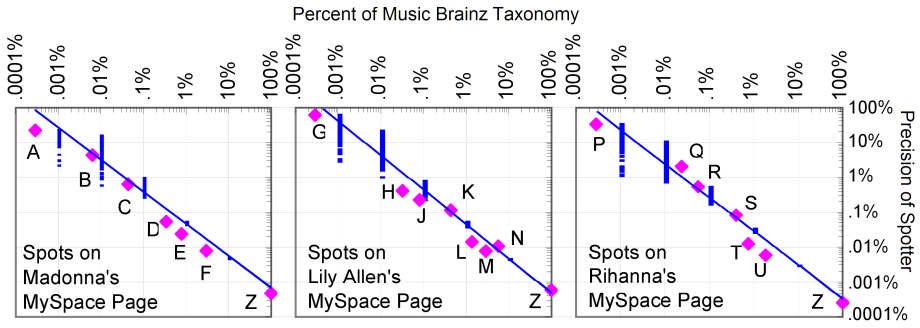


Fig. 4. Naive spotter using selected portions of the MusicBrainz RDF based on descriptive characteristics of Madonna, Lily Allen and Rihanna, respectively. The Key to the data points is provided in Table 4

The agreement of the three types of restrictions from above with the random restrictions from the previous Section are clear from comparing the plots in Figure 4. This confirms the general effectiveness of limiting domain size to improve precision of the spotter, regardless of the type of restriction, as long as the restriction only removes off-target artists. A reduction in the size of the RDF graph results in an approximately proportionate increase in precision.

This is a particularly useful finding, because it means that any restriction we can apply will improve precision, and furthermore we can estimate the improvement in precision.

5 NLP Assist

While reducing extraneous possibilities improved precision of the naive spotter significantly, false positives resulting from spots that appear in different senses still need attention (see Table 5). The widely accepted *'one sense per discourse'* notion[17] that the sense or meaning of a word is consistent within a discourse does not hold for this data given the use of common words as names for songs and albums.

The task is to assess whether a spot found is indeed a valid track or album. This is similar to the word sense disambiguation problem where the task is to resolve which one of many pre-defined senses is applicable to a word[9]. Here, we use a learning algorithm over local and surrounding word contexts, an approach similar in principle to several past efforts but adapted to our domain of discourse [16].

Valid: Got your new album *Smile*. Loved it!
Invalid: Keep your *SMILE* on. You'll do great!

Table 5. Spots in multiple senses

Formally, our task can be regarded as a binary classification problem. Consider the set of all spots found by the naive spotter. Each spot in this set can be labeled 1 if it is a track; and -1 if it is not, where the label is associated with a set of input features that characterize a spot s. This is implemented as

a Support Vector Machine (SVM), a machine learning approach known to be effective for solving binary pattern recognition, named entity recognition and document classification problems[11].

5.1 Features

We trained and tested the SVM learner on two sets of features collectively observed in the tagged data (see Section 3.1); *basic features*, that characterize a spot and *advanced features* that are based on the context surrounding the spot.

Basic features: We encode a set of spot-level boolean features (see Table 6) that include whether the spot is all capitalized, starts with capital letters or is enclosed in quotes. If the entire comment including the spot is capitalized, we do not record a 1 for *s.allCaps* or *s.firstCaps*. We also encode features derived from the part-of-speech (POS) tags and NP-chunking of comments (see syntactic features in Table 6)[6]. To encode syntactic features, we created a list of the Penn Treebank tag set[7] also used by the Stanford parser. If the parser returns a tag for the spot, we obtain the tag's index position in the list to encode this feature. If the sentence is not parsed this feature is not encoded.

Advanced features: We encode the following advanced features intended to exploit the local context surrounding every spot. We encode the POS tags of word tokens appearing before and after a spot in a sentence.
Sentiment expressions and domain-specific terms: We found that spots that co-occurred with sentiment expressions and domain-specific words such as 'music', 'album', 'song', 'concert', etc. were more likely to be valid spots. We encode these boolean features in the following manner.

First, we curated a sentiment dictionary of 300 positive and negative expressions from UrbanDictionary [8] (UD), given the use of slang by this poster demographic. Starting with expressions such as 'good', and 'bad', we obtained the top 10 related sentiment expressions for these words. We continued this process for the newly obtained words until we found no new words. Note that we are not concerned with the polarity, but mere co-occurrence of sentiment expressions with spots. A dictionary of 25 domain-specific terms, such as 'music', 'album', 'track', 'song' etc. was created manually by consulting MusicBrainz. These dictionaries are available for download from the Knoesis Center website[9].

If one or more sentiment expressions, domain-specific terms or their word forms were spotted in the same sentence as the spot, values for $s.S_{sent}$ and $s.S_{dom}$ are recorded as 1. Corresponding $s.C_{sent}$ and $s.C_{dom}$ features were also used to record similar values when these terms were found elsewhere in the comment. Encoding the actual number of co-occurring sentiment or domain expressions did not significantly change the classification result.

[6] Obtained using the Stanford NL Parser http://nlp.stanford.edu/software/lex-parser.shtml
[7] http://www.cis.upenn.edu/~treebank/
[8] www.urbandictionary.com
[9] http://knoesis.wright.edu/research/semweb/music

Syntactic features	**Notation-S**
$^{+}$POS tag of s	s.POS
POS tag of one token before s	s.POS$_b$
POS tag of one token after s	s.POS$_a$
Typed dependency between s and sentiment word	s.POS-TD$_{sent}{}^{*}$
Typed dependency between s and domain-specific term	s.POS-TD$_{dom}{}^{*}$
Boolean Typed dependency between s and sentiment	s.B-TD$_{sent}{}^{*}$
Boolean Typed dependency between s and domain-specific term	s.B-TD$_{dom}{}^{*}$
Word-level features	**Notation-W**
$^{+}$Capitalization of spot s	s.allCaps
$^{+}$Capitalization of first letter of s	s.firstCaps
^{+}s in Quotes	s.inQuotes
Domain-specific features	**Notation-D**
Sentiment expression in the same sentence as s	s.S$_{sent}$
Sentiment expression elsewhere in the comment	s.C$_{sent}$
Domain-related term in the same sentence as s	s.S$_{dom}$
Domain-related term elsewhere in the comment	s.C$_{dom}$

$^{+}$Refers to basic features, others are advanced features

*These features apply only to one-word-long spots.

Table 6. Features used by the SVM learner

Typed Dependencies:

We also captured the typed dependency paths (grammatical relations) via the *s.POS-TD$_{sent}$* and *s.POS-TD$_{dom}$* boolean features. These were obtained between a spot and co-occurring sentiment and domain-specific words by the Stanford parser[12] (see example in 7). We also encode a boolean value indicating whether a relation was found at all using the *s.B-TD$_{sent}$* and *s.B-TD$_{dom}$* features. This allows us to accommodate parse errors given the informal and often non-grammatical English in this corpus.

Valid spot: Got your new album **Smile**. Simply *loved* it!
Encoding: nsubj(loved-8, Smile-5) implying that **Smile** is the nominal subject of the expression *loved*.

Invalid spot: Keep your **smile** on. You'll do *great*!
Encoding: No typed dependency between **smile** and *great*

Table 7. Typed Dependencies Example

5.2 Data and Experiments

Our training and test data sets were obtained from the hand-tagged data (see Table 3). Positive and negative *training examples* were all spots that all four annotators had confirmed as valid or invalid respectively, for a total of 571 positive and 864 negative examples. Of these, we used 550 positive and 550 negative examples for training. The remaining spots were used for test purposes.

Our positive and negative *test sets* comprised of all spots that three annotators had confirmed as valid or invalid spots, i.e. had a 75% agreement. We also included spots where 50% of the annotators had agreement on the validity of the

	Features	Valid	Invalid Spots			Acc.
		Set1	Set2	Set3	Avg.	Split
(1)	W	45	88	84	**86**	45 - 86
(2)	W+S	74	43	37	40	74 - 40
(3)	W+D	62	85	83	**84**	62 - 84
(4)	D	70	50	62	56	70 - 56
(5)	D+S	72	34	36	35	72 - 35
(6)	W+D+s.POS	61	66	74	70	61 - 70
(7)	W+D+s.POS$_{b,a}$+s.POS-TDs	**78**	47	53	50	78 - 50
(8)	W+D+s.POS$_{b,a}$+s.B-TDs	**90**	33	37	35	90 - 35
(9)	W+D+only s.POS$_{b,a}$	62	81	87	**84**	62 - 84
(10)	W+D+only s.POS-TDs	60	79	91	**85**	60 - 85
(11)	W+D+only s.B-TDs	71	68	72	70	71 - 70
(12)	All features	42	89	93	**91**	42 - 91

Table 8. Classifier accuracy in percentages for different feature combinations. Best performers in bold.

spot and the other two were *not sure*. We further divided our negative test set into two disjoint equal sets that allowed us to confirm generality of the effect of our features. Finally, our test set of *valid spots*, Set 1, contained 120 spots and the two test sets for *invalid spots*, Set 2 and Set 3, comprised of 229 spots each.

We evaluated the efficacy of features shown in Table 6 in *predicting the labels assigned by the annotators*. All our experiments were carried out using the SVM classifier from [5] using 5-fold cross-validation. As one way of measuring the relative contribution of advanced contextual and basic spot-level features, we removed them one after another, trying several combinations. Table 8 reports those combinations for which the accuracy in labeling either the valid or invalid datasets was at least 50% (random labeling baseline). Accuracy in labeling valid and invalid spots refer to the percentage of true and false positives that were labeled correctly by the classifier. In the following discussion, we refer to the average performance of the classifier on the false positives, Sets 2 and 3 and its performance on the true positives, Set 1.

5.3 Usefulness of Feature Combinations

Our experiments revealed some expected and some surprising findings about the usefulness of feature combinations for this data. For valid spots, we found that the best feature combination was the word-level, domain-specific and contextual syntactic tags (POS tags of tokens before and after the spot) when used with the boolean typed dependency features. This feature combination labeled 90% of good spots accurately. The next best and similar combination of word-level, domain-specific and contextual tags when used with the POS tags for the typed dependency features yielded an accuracy of 78%. This suggests that *local word descriptors along with contextual features* are good predictors of valid spots in this domain.

For the invalid spots (see column listing average accuracy), the use of all features labeled 91% of the spots correctly. Other promising combinations included

the word-level; word-level and domain-specific; word-level, domain-specific and POS tags of words before and after the spot; word-level, domain-specific and the typed dependency POS tags, all yielding accuracies around 85%.

It is interesting to note that the POS tags of the spot itself were not good predictors for either the valid or invalid spots. However, the POS typed dependencies were more useful than the boolean typed dependencies for the invalid spots. *This suggests that not all syntactic features are useless, contrary to the general belief that syntactic features tend to be too noisy to be beneficial in informal text.* Our current investigations to improve performance include the use of other contextual features like commonly used bi-grams and tri-grams and syntactic features of more tokens surrounding a spot.

Feature Combination	Mean Acc.	Std.Dev Acc.
All Features	99.4%	0.87%
W	91.3%	2.58%
W+D	83%	2.66%
W+D+only s.POS-TDs	80.8%	2.4%
W+D+only s.POS$_{b,a}$	77.33%	3.38%

Table 9. Average performance of best feature combinations on 6 sets of 500 invalid spots each

Accuracy in Labeling Invalid Spots: As further confirmation of the generality of effect of the features for identifying incorrect spots made by the naive spotter, we picked the best performing feature combinations from Table 8 and tested them on a dataset of 3000 known invalid spots for artist Rihanna's comments from her MySpace page. This dataset of invalid spots was obtained using the entire MusicBrainz taxonomy excluding Rihanna's song/track entries - effectively allowing the naive spotter to mark all invalid spots. We further split the 3000 spots into 6 sets of 500 spots each. The best feature combinations were tested on the model learned from the same training set as our last experiment. Table 9 shows the average and standard deviation performance of the feature combinations across the 6 sets. As we see, the feature combinations performed remarkably consistently for this larger test set. The combination of all features was the most useful, labeling 99.4% of the invalid spots correctly.

6 Improving Spotter Accuracy Using NLP Analysis

The last set of experiments confirmed the usefulness of the features in classifying whether a spot was indeed a track or not. In this next experiment, we sought to measure the improvement in the overall spotting accuracy - first annotating comments using the naive spotter, followed by the NLP analytics. This approach of boosting allows the more time-intensive NLP analytics to run on less than the full set of input data, as well as giving us a certain amount of control over the precision and recall of the final result.

Figure 5 shows the improvement in precision for spots in the three artists after boosting the naive spotter with the NLP component. Ordered by decreasing recall, we see an increase in precision for the different feature combinations. For

example, the precision of the naive spotter for artist Madonna's spots was 23% and almost 60% after boosting with the NLP component and using the feature combinations that resulted in a $42-91$ split in accurately labeling the valid and invalid spots.

Although our classifier was built over the results of the naive spotter, i.e. it already knew that the spot was a potential entity, our experiments suggest that the features employed might also be useful for the traditional named entity recognition problem of labeling word sequences as entities.

Our experiments also suggest that although informal text has different characteristics than formal text such as news or scientific articles, simple and inexpensive learners built over a dictionary-based naive spotter can yield reasonable performance in accurately extracting entity mentions.

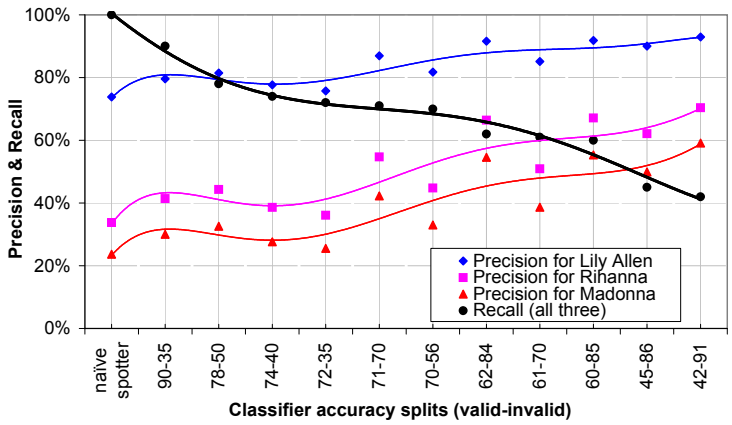


Fig. 5. NLP Precision-Recall curves for three artists and feature combinations

7 Conclusion and Future Work

Spotting music tracks in Informal English is a critical enabling technology for applications such as the BBC SoundIndex that allows real-time tracking of opinions in on-line forums. The presented approach is applicable to other domains as well. We are currently adopting our system to spot automobiles and find that car brands, makes and models provide a well formed ontology as well. We believe that such on-demand information will play an increasingly important role in business as companies seek to better understand their customers. Rapid detection of events (e.g. the artist "OK Go" ramping up the chart within hours of being featured on the popular TV show Big Brother) illustrate the possibilities of these systems.

There are several challenges in constructing these systems. Discussions of music produce millions of posts a day, which need to be processed in real-time, prohibiting more computational intensive NLP techniques. Moreover, since 1920,

song titles based on common words or phrases have become very popular (see Figure 3), making it difficult to spot and disambiguate song titles.

In this paper, we presented a two stage approach - entity spotting based on scoping a domain model followed by SVM based NLP system to facilitate higher quality entity extraction. We studied the impact of restricting the size of the entity set being matched and noted that the spot frequency follows a Zipf distribution. We found that R^2 for this distribution is fairly consistent among a sample of artists. This allows a reasonable a priori evaluation of the efficacy of various restriction techniques using the calibration curve shown in Figure 2. We found that in many cases such restrictions can come from the language of the spotted text itself.

Given these potential spots, we show that simple classifiers trained on generic lexical, word and domain specific characteristics of a spot can effectively eliminate false positives in a manner that can improve accuracy up to a further 50%. Our experiments suggest that although informal text has different characteristics than formal text, learners that improve a dictionary-based naive spotter can yield reasonable performance in accurately extracting entity mentions.

7.1 Future Work

Future areas of interest include applying standard measures such as TF-IDF to predict the ambiguity of entities for use with the NLP component. One drawback of the current approach for scoping the linked data or RDF graph to a single artist occurs when references to multiple artists are made in text (e.g. your song "X" reminds me of Y's song "Z"). Even though these mentions are sparse, we are planning to include non-ambiguous artist names as "activators" in the base spotting set. If a post mentions another artist, the spotter would temporarily activate entities from the RDF belonging to that specific artist.

Another area of interest is to examine automatic constraint selection based on the posts themselves. For example a "birthdate note" detector, a gender of artist identifier, a recent album release detector, etc. Using the Zipf distribution in Figure 2 we can estimate how helpful each detector might be before we implement it. Once a robust set of post based constraint detectors are developed we can begin to experiment on "free domain" spotting - that is spotting in domains where less focused discussions are expected, e.g. Twitter messages.

We also plan to extend our work to other free text domains. The ability to achieve reasonable performance in this problem suggests that this approach will work well in other, less challenging domains where the entities are less overlapping (e.g. company name extraction) or the English is less informal (e.g. news releases).

8 Acknowledgements

We would like to thank the BBC SoundIndex team for their help and support. We would also like to thank Marti Hearst, Alexandre Evfimievski, Guozhu Dong,

Amy Vandiver and our shepherd Peter Mika for their insightful reviews and suggestions.

References

1. A. Alba, V. Bhagwan, J. Grace, D. Gruhl, K. Haas, M. Nagarajan, J. Pieper, C. Robson, and N. Sahoo. Applications of voting theory to information mashups. In *ICSC*, pages 10–17. IEEE Computer Society, 2008.
2. R. Ananthanarayanan, V. Chenthamarakshan, P. M. Deshpande, and R. Krishnapuram. Rule based synonyms for entity extraction from noisy text. In *ACM Workshop on Analytics for noisy unstructured text data*, pages 31–38, 2008.
3. E. Bozsak, M. Ehrig, S. Handschuh, S. H, A. Maedche, A. Hotho, E. Maedche, B. Motik, D. Oberle, C. Schmitz, N. Stojanovic, Rudi, S. Staab, L. Stojanovic, and V. Zacharias. Kaon - towards a large scale semantic web. In *Proc. of EC-Web 2002, LNCS*, pages 304–313. Springer, 2002.
4. R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics, 2006.
5. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
6. H. L. Chieu and H. T. Ng. Named entity recognition: A maximum entropy approach using global information. In *COLING*, 2002.
7. C. Cunha, A. Bestavros, and M. Crovella. Characteristics of www client-based traces. Technical report, Boston University, Boston, MA, USA, 1995.
8. O. Etzioni, M. Cafarella, and et al. Web-scale information extraction in knowitall: (preliminary results). In *WWW '04*, pages 100–110. ACM, 2004.
9. N. Ide and J. Vronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24:1–40, 1998.
10. B. A.-M. J Hassell and I. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. In *Proceedings of the International Semantic Web Conference, ISWC*, 2006.
11. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Lecture Notes in Computer Science: Machine Learning*, pages 137–142. Springer Verlag, 1998.
12. M. Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pages 449–454, 2006.
13. E. Minkov, R. C. Wang, and W. W. Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *HLT/EMNLP*. The Association for Computational Linguistics, 2005.
14. C. Muller and I. Gurevych. Using wikipedia and wiktionary in domain-specific information retrieval. In *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark, 2008.
15. D. Nadeau and S. Sekine. *A survey of named entity recognition and classification.* Linguisticae Investigationes, 2007.
16. D. Tatar. Word sense disambiguation by machine learning approach: A short survey. *Fundam. Inf.*, 64(1-4):433–442, 2004.
17. D. Yarowsky. *Hierarchical Decision Lists for WSD.* Kluwer Acadmic Publishers, 1999.
18. G. K. Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology.* Addison-Wesley, Cambridge, Mass, 1949.