FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO


FEUP

# Inferring semantics from lyrics using weakly annotated data

**Diogo Costa**

Master in Informatics and Computing Engineering

Supervisor: Fabien Gouyon (Ph.D.)

Co-Supervisor: Luís Sarmento (Ph.D.)

23$^{rd}$ January, 2012

# Inferring semantics from lyrics using weakly annotated data

**Diogo Costa**

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: João Moreira (Ph.D.)
External Examiner: Paulo Novais (Ph.D.)
Supervisor: Fabien Gouyon (Ph.D.)

---

$8^{th}$ February, 2012

# Abstract

Human annotated data is often the starting point in modeling and evaluating automatic text categorization approaches. However, the quality of this data might not always be ideal, which includes but is not restricted to, partial annotations. Therefore understanding the implications of treating such data as golden standards for text categorization could prove to be useful.

In this thesis we will measure the impact of partially annotated data when modeling and evaluating a predictive model for automatic lyrics categorization representing its semantic content, using traditional text categorization methodologies and a partially annotated dataset which we will expand with external sources of annotations.

We will show that the impact of partially annotated data is more prominent when evaluating a predictive model than in its modeling phase. We will also superimpose these results with other types of textual data and argue lyrics categorization is a particularly hard categorization problem.

# Resumo

Dados manualmente anotados são muitas vezes o ponto de partida para a modelação e avaliação de abordagens de categorização de texto automático. No entanto a qualidade destes dados não são sempre as ideais, isto inclui mas não se restringe a, dados parcialmente anotados. Compreender as implicações de usar este tipo de dados como "padrão de ouro" pode portanto ser útil para a categorização de texto.

Nesta tese iremos medir o impacto de dados parcialmente anotados quando modelamos e avaliamos um modelo preditivo para categorização automática de letras de musica em tópicos representativos do seu conteúdo semântico, usando abordagens tradicionais de categorização de texto e dados parcialmente anotados que posteriormente serão expandidos com anotações de outras fontes.

Mostraremos que o impacto deste tipo de dados é mais notório na fase de avaliação do modelo preditivo do que na fase de modelação. Iremos também sobrepor os nossos resultados com outros tipos de dados textuais e argumentaremos que a tarefa de categorização de letras musicais é um caso particularmente difícil de categorização.

*"When we try to pick out anything by itself we find that it is bound fast by a thousand invisible cords that cannot be broken, to everything in the universe."*

John Muir (1869)

# Contents

# List of Figures

LIST OF FIGURES

# List of Tables

LIST OF TABLES

# Abbreviations

| | |
|---|---|
| MIR | Music Information Retrieval |
| IR | Information Retrieval |
| NLP | Natural Language Processing |
| TP | True positive |
| TN | True negative |
| FP | False positive |
| FN | False negative |
| BEP | Break-even Point |
| SVM | Support Vector Machines |

# Chapter 1

# Introduction

In this chapter we will cover the motivation behind this thesis, a broad overview of related work and what will be our contribution.

## 1.1 Motivation

In recent years attention on songs lyrics as gained momentum in emotion identification [YLC+08, YL09] automatic indexing and retrieval [LKM04, MMC+05] and classification of music [WZO07, KKP08, MNR08] not only as a standalone procedure but also as a complement to the traditional acoustic analysis as [YLC+08, LKM04] points out.

As [BCD03] points out, in 2003, over 30% of Music Information Retrieval (MIR) on the web is made using a fragment of a songs lyrics or the lyrics story. This is not the only evidence of a growing interest in lyrics [Ven11].

Advances in automatic categorization of song lyrics are therefore pertinent to MIR and Information Retrieval (IR) overall.

Having access to a set of human annotated song lyrics is crucial in developing such an approach to lyrics categorization, and indeed access to annotated data is possible through closed and open services on the web. However this type of annotations can, and often is, prone to a number of problems that may undermine such an approach.

By analyzing a song lyric the annotator identifies, to the best of his abilities, its defining characteristics which in turn defines membership within one or more abstract concepts. He then chooses the natural language terms, the labels, that best identify those abstract concepts and uses them accordingly for annotation.

For this annotation to be perfect several rules must apply. Firstly the annotator must have complete knowledge over the object he is annotating and over the labels that he may use. This is often not the case and originates several types of problems in our annotated data. Data which suffers from these problems will be referred henceforth as weakly annotated data.

One of the problems in weakly annotated data are partial annotations, in which a documents annotation consists only of a subset of its unknown "true" annotation. Our thesis will focus on this problem alone.

We should mention however some of the other problems associated with weakly annotated data like the use of polysemous, synonymous, misspelled or faceted labels. Inconsistent annotations are also a problem since two human annotators might not agree on the correct annotation for a given song lyric.

Understanding the impact of such shortcomings could be helpful in developing approaches to automatically categorize not only song lyrics and poetic text but also other types of data which contain partially annotated data.

## 1.2 Related Work

Work done in text categorization of lyrics include [MMC+05] which categorizes songs into five different topics (Love, Violent, Protest (antiwar), Christian and Drugs). Although these topics are very course grained, they show how using common text categorization techniques can be successfully applied to lyrics.

However some observations regarding the difficulty of natural language processing of lyrics have been raised [WZO07], due to "heavy use of metaphoric, rhyming, unusual and archaic language" [SM98]..

Weakly annotated data has also been discussed and tackled previously by [LSM10] which shows how to deal with synonymous and misspelled labels in annotated acoustic data. They use a mixture of supervised and unsupervised approach and reveal that performance measures using this kind of data grossly underestimate the performance of such approaches.

## 1.3 Contribution

In this thesis we will measure the impact of weakly annotated data when fitting and evaluating a predictive model for automatic lyrics categorization.

More specifically we will focus on using a predictive model that categorizes lyrics into topics representing its semantic content and evaluating how using partially annotated data affects such a model, using traditional IR measures.

We will accomplish this using traditional text categorization methodologies and a partially annotated dataset which we will expand with external sources of annotations.

## 1.4 Outline

**Chapter 2** goes into greater detail of some of the related work that tackles connected or similar problems.

**Chapter 3** formalizes what are partially annotated data and introduces a specific dataset.

**Chapter 4** clearly identifies our approach by defining methodologies and evaluation procedures.

**Chapter 5** will present and discuss the results of the proposed approach.

**Chapter 6** provides a broad description of the thesis and if the objectives were achieved.

# Chapter 2

# Related work

In this chapter we will review related work by other authors. We will summarize and make a critic analysis of several similar or connected problems.

## 2.1 Text categorization

State of the art text categorization approaches have been made using supervised approaches [MN98, Joa98, SM98, SM99], semi-supervised [NMTM00, Joa99], unsupervised [Hof99, BNJ03, XLG03, MLM07] or combinations of these learning algorithms [RHNM09].

Supervised and semi-supervised approaches take as input labeled data (and in semi-supervised unlabeled data) and output predictions of labels for new data.

Although these approaches have become quite successful, unsupervised approaches are becoming more popular in recent years. These do not depend on labeled data and are essentially cluster algorithms which group documents of text into non-observable high-abstraction concepts usually representing strong co-occurrence of words.

Both high-abstraction concepts as well as concrete labels are important in an information retrieval context. If we wish to search for similar documents of text we might opt to use a unsupervised approach, whereas if we wish to search for documents that belong to a concept identified by a specific label we would use a supervised or semi-supervised approach.

## 2.2 Lyrics categorization

We will now focus our attention to previous work done in text categorization using song lyrics.

Work done by [MMC+05] include a supervised algorithm [MN98] for categorization of song lyrics into five different topics (Love, Violent, Protest (antiwar), Christian

and Drugs) representing its semantic content. It is difficult to evaluate their performance because the authors only provide a number of an unknown measure. Also Since there are few, very different, categories it was also unclear how this classifier would escalate with more fine-grained categories. Regardless, this is the only work, as far as this author knows, that directly relates to the problem of text categorization of song lyrics.

An example of unsupervised lyrics categorization is [LKM04]. Although the objective of the authors is to find similar artists, by clustering them, they use the songs lyrics to achieve those clusters using word co-occurrence. The authors therefore assume that an artist may be characterized by the semantic contents of its lyrics.

They first apply topic modeling [Hof99] to a lyrics dataset to obtain a set of latent topics. They then project each artist in those latent topics by using the words which occur in his lyrics. These projection vectors are then used to compute similarity using the $L_1$ norm ($\sum_{i=1}^{n} |x_i|$).

Their evaluation procedure used a survey of actual users that picked similar artists, and compared their approach with random guessing and state of the art acoustic similarity. Overall the acoustic metric was the best of the three, and the difference between the proposed approach and random guessing was not impressive. The authors justify this by saying the users tend to pick artists that sound the similar instead of artists that sing about related topics, which is what they are doing.

However we argue that topic modeling could indeed help with text categorization of semantic content.

A different unsupervised approach to lyrics categorization is the keyword generation approach [WZO07]. Their objective is to generate words that best identify the songs semantic content. The authors argue that "...topic words in lyrics have a very small number of occurrences. Instead a large portion will be devoted to the background" [WZO07].

Their proposed solution works in two steps, first they cluster similar sentences and for each cluster find the 10 most representative words which are expanded using WordNet [1]. In the second step they choose the cluster that has the most shared keywords with other groups of sentences in other lyrics because they argue the main topic is shared throughout different lyrics while the background is not. The authors then compare their approach to previous results on the same ground-truth using a supervised approach and claim to have a lower error rate.

More importantly their approach is the first to acknowledge some of the difficulties of text categorization when applied to song lyrics.

[1] `http://wordnet.princeton.edu/`

## 2.3 Dealing with weakly annotated data

We have already mentioned the characteristics of weak labeling and particularly that they might have an impact on building predictive models for automatic text categorization. Their impact however is not restricted to text categorization.

The objective of work done in [LSM10] is to generate tags for music using weakly annotated acoustic data. The authors argue these tags "can be used to support music search and recommendation on a semantic level." but also argue about the noisiness in tags made by users.

The authors approach is to map songs into a "well behaved" latent concept space via topic modeling, and then proceed categorization using this concept space. By using tag co-ocurrence, these concepts encapsulate synonymous, misspelled or semantically close tags in the form of a probability distribution over all the tags $p(t|z)$. Notice the number of concepts can be much smaller than the original tag space.

Song $x$ is now modeled in this abstract concept space and not in the original tag space producing $p(z|s)$.

To infer the tags for a new song they simply combine both predictive models to generate a posterior probability containing the distribution over the tag space for that song (Equation 2.1).

$$p(t = j|s = i) = \sum_k p(t = j|z = k)p(z = k|s = i) \tag{2.1}$$

Although this approach has similar accuracy to more common supervised approaches, it is much more efficient.

It is interesting to see that using only the initial ground-truth as gold standard grossly underestimates the approach performance, simply because tags that are correctly predicted will be marked as wrong predictions if in the ground-truth those labels aren't present.

This is further analyzed by asking humans to evaluate the predictions, and of the tags marked as correct by human evaluators roughly 50% weren't in the ground-truth. This is clearly a problem of partially annotated data.

In [SGO09], the authors present a novel method on how to add more annotations to partially annotated data using a specific external source of data. Although their motivation is not text categorization, the method used could be helpful in measuring the annotations quality of the original data.

The authors first start with artists tags scraped from Last.fm, then for each artist that has a Wikipedia article, words from that article are extracted and only words that are used as tags from Last.fm are selected. Each selected word is ranked using a weighting scheme and the top words that are not already used in the Last.fm tags are assigned to the artist.

# Chapter 3

# Data

## 3.1 Partially annotated data

We will define text categorization as the assignment of an arbitrary number of labels from set $L = \{L_1, L_2, ..., L_n\}$, to document $D$. The set of labels assigned to document $D$ forms its annotation $A$.

This task can, and often is, performed by a human annotator. We represent the annotator as function $Y(D) = A$, where $A \subseteq L$.

Not only can $L$ potentially contain synonymous or polysemous labels but it can also contain labels which could describe $D$ in different ways.

Annotation $A$ is also influenced of individual cognitive and intellectual limitations of the annotators. The quality of annotations are bounded both by the expertise of the annotator and by the complexity of the categorization task. For example, if $L$ is sufficiently large most annotators know only a subset of all possible labels. This is specially true in categorization using open vocabularies.

Furthermore, if different annotators perceive $D$ in different ways, can we always find a consensus between them to establish the "true" annotation $A$?

One result of these shortcomings is partially annotated data, in which $A$ partially describes $D$. In other words, if an oracle would tell us the perfect annotation $A_o$ then, at best, $A \subseteq A_o$. We know only the true positives for each category and cannot separate true from false negatives.

We wish to estimate $Y$ based on its observable outputs $A_i$ for several $D_i$. Such a function would be $Y^*(D) = A^*$, where $A^* \subseteq L$. Are partial annotations sufficient to estimate such a function? Even if we naively assume that perfect annotation is possible and that we can create a function which produces perfect annotations, how do we measure its success if we only have partially annotated data to test it against?

## 3.2 Annotated lyrics case study

Our data was collected by Mohamed Sordo [1] from Pompeu Fabra University [2].

The data consists of 17795, English written, song lyrics fetched from MusixMatch [3] and expert annotations fetched from Songfacts [4].

For some lyrics these annotations contain information about its semantic content (topics) as labels. To avoid problems of synonymy and too much fine grained labels we have manually grouped the initial 104 labels into 50 labels.

Of all the lyrics only 39% have at least one annotation, from those 96% have only one annotation and there is no example of a song annotated with more than three categories. Because the actual process of annotation by Songfacts didn't use a taxonomy or any type of schema, we will consider this as expertly tagged data, subject to problems of weakly annotated data (Section **??**).

### 3.2.1 Lyrics

As an example Table 3.1 shows an excerpt from a particular song lyrics.

| | |
|---|---|
| Lights go out and I can't be saved | Singin', come out upon my seas |
| Tides that I tried to swim against | Curse missed opportunities |
| Brought me down upon my knees | Am I, a part of the cure |
| Oh I beg, I beg and plead | Or am I part of the disease |
| ... | ... |
| Confusion never stops | Home, home, where I wanted to go |
| Closing walls and ticking clocks | Home, home, where I wanted to go |
| Gonna, come back and take you home | Home, home, where I wanted to go |
| I could not stop that you now know | Home, home, where I wanted to go |

Table 3.1: Example of song named "Clocks"

Some sentences appear to be ill defined ("Yooooooooooooo ohhhhhh " ) and there is extensive use of allegory ("...come out upon my seas"). Using the title and the text of the lyrics we can suggest some possible categories like "adversity" ("Lights go out and I can't be saved"), "time" ("Closing walls and ticking clocks") or someone who is "homesick" ("Home, home, where I wanted to go").

The five most frequent terms - "ohhhhhh" (10), "home" (9), "yooooooooooooo" (8), "go" (5) and "where" (4) - don't seem to summarize well any relevant topic.

We have found the lyrics vocabulary in the data roughly follow Zipfs law.

[1] http://www.dtic.upf.edu/~msordo/
[2] http://www.upf.edu/en/
[3] http://musixmatch.com/
[4] http://www.songfacts.com/

### 3.2.2 Annotations

Like we mentioned before, the annotations represent the semantic content of the lyrics, its topic.

According to human experts the song in Table 3.1 is about "...being in a conflicted, but very intense relationship as precious time slips away" and "...the world's obsession with time". This is an example of how cryptic the content of lyrics can be.

Figure 3.1 shows the category distribution which has an entropy of 5.12 (bits), if all the categories were evenly distributed we would have an entropy of 5.64 (bits), this simply means that our distribution is somewhat skewed towards few categories.

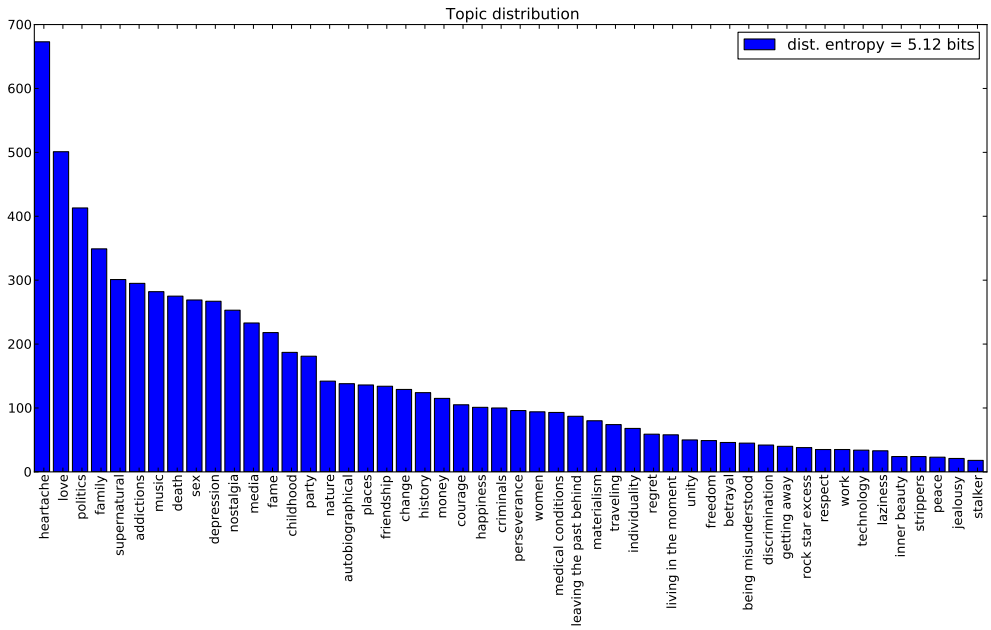


Figure 3.1: Topic distribution.

### 3.2.3 Dependency between lyrics and annotations

As we've mentioned the data contains song lyrics along with some annotations about its semantic content, its topic.

We argue that this type of text is a particularly hard case for natural language processing, and is due to the ambiguous nature of poetic texts which can include, but are not restricted to, allegories or polysemous words.

We will test the hypothesis of independence between each word and categories "religion" and "politics" for two different types of text, poetic and non-poetic text.

We will use the lyrics from our dataset with respective tags extracted from Last.fm and the 20 NewsGroups dataset [5]. Both datasets contain text from categories "religion" and "politics", so we may superimpose results of feature selection.

To test our null hypothesis we have used the $\chi^2$ test to compute the p-values for every word and each category.

In Figure 3.2 we have plotted the $1 - p\text{–}value$ for every word for both lyrics and newsgroups text for category "religion". In lyrics very few words show a dependency of the category only 0.9% of the words have a p-value equal or less than 0.05. In contrast using text from newsgroups, in general, a large part of words do show evidence of dependency, 10% to be more precise, an increase by a factor of 10.

Figure 3.3 shows the same approach for category "politics". Similarly to category "religion" words from lyrics seem to be much more independent of the category, in this case only 3% of words have a p-value equal or less than 0.05, while in the newsgroups case this number is 17%.

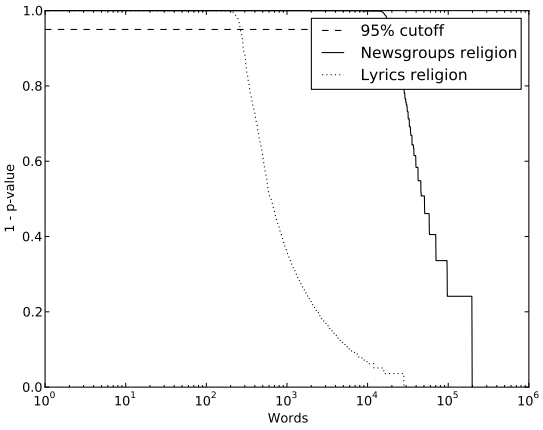


Figure 3.2: Statistical evidence of independence between words and category "religion" for different types of text.

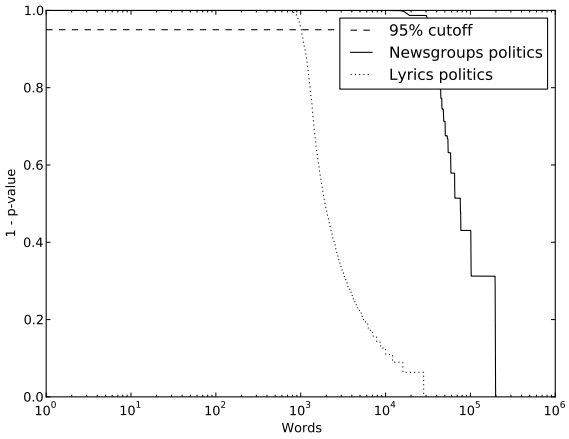


Figure 3.3: Statistical evidence of independence between words and category "politics" for different types of text.

Table 3.2 shows some of the words with the lowest p-value for both chosen categories.

[5] http://people.csail.mit.edu/jrennie/20Newsgroups/

| | Religion | Politics |
|---|---|---|
| Lyrics | thieves, prayer, cuz', christian, prayin', sinnerman, lord , power, temple, jews, boilin' | justice, crew, fox, naaah, li, ren, elected, gitalong |
| Newsgroups | life, ax, jesus, god, lord, sin, biblical, doctrine, mary, scripture, love, christians, elohim, christian, believe, christ, father, clh, rutgers, athos, homosexuality, catholic, son, truth, bible, eternal, holy, paul, jehovah, testament, spirit, marriage, resurrection, faith, church, heaven, sins, christianity | fbi, american, armenia, israel, ax, arab, killed, mr, country, clinton, arabs, optilink, president, cramer, muslims, state, azerbaijan, stratus, myers, serdar, waco, military, fire, villages, people, sera, atf, soldiers, said, palestinian, turks, turkish, gun, police, policy, turkey, states, jewish, russian, rights, crime, guns, zuma, muslim, israeli, armenian, jews, children, argic, batf, armenians, troops, genocide, weapons, stephanopoulos, government, war, adl, population, firearms, palestinians, clayton, arms, slut, clampdown |

Table 3.2: Words with the lowest p-value for categories "religion" and "politics" in song lyrics and newsgroups text.

Data

# Chapter 4

# Experimental setup

In this chapter we will define the approach to achieve our objective of evaluating the impact of weakly annotated data in text categorization of lyrics.

This chapter specifies the methodologies used in evaluating the impact of partially annotated data in lyrics categorization. We will first define the approach used to expand our initial dataset with third party annotations (Section 4.1), the methodology used in lyrics categorization (Section 4.2), and then define the methodology used to test the impact of partially annotated data (Section 4.3).

## 4.1 Data expansion

We will now detail the process we used to expand the annotations of our initial data.

Human generated meta-data, about lyrics, is available through websites like Songfacts, Million Song dataset [1] or Allmusic [2].

By using such services we can extract human made content about the songs which may include a snippet of text regarding the songs semantic content. Because in some cases this type of meta-data is provided in natural language, the task of extracting the important information could potentially be as hard as our original problem.

Like Songfacts, Last.fm [3] provides easy access to songs meta-data in the form of tags created by users in their collaborative tagging system. However, unlike the Songfacts, these tags have no clear meaning to what they represent.

We have used a very naive approach to complement our data with this source of annotations. For every song lyrics we fetch the tags associated with it on Last.fm (Step 1 in Figure 4.1). If the song contains tags then we go trough each tag and if it is equal to any of the 50 initial labels and that song hasn't been annotated with it, we add the tag as a new label for that songs lyrics (Step 2 in Figure 4.1).

[1] http://labrosa.ee.columbia.edu/millionsong/

[2] http://www.allmusic.com/

[3] http://www.last.fm/

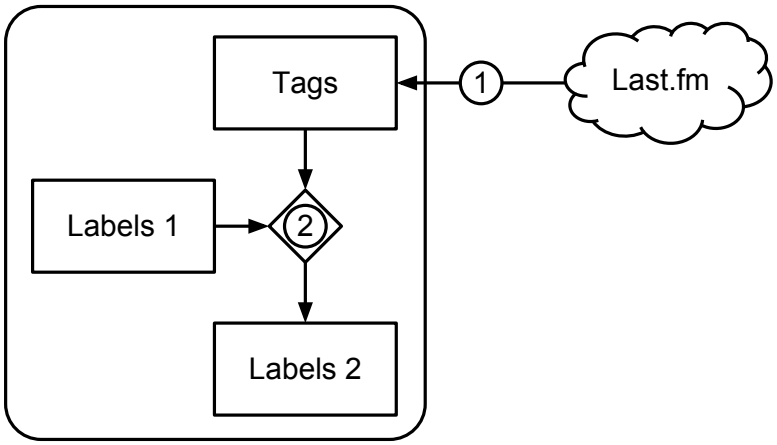


Figure 4.1: Re-annotation process. 1) We fetch a songs tags. 2) We include tags that belong to our 50 label vocabulary and have not been used in annotating that song.

After the reannotation process, each song lyric contains a set of annotations from Last.fm (Labels 2) as well as a set of annotations from our initial data (Labels1)

Having done this annotation consider now Figure 4.2 which divides our data into two sets, set $S$ of song lyrics with annotations from our initial data (Labels1) and the set $T$ of song lyrics with annotations from our re-annotation process (Labels2). $D$ represents the set of all lyrics in the dataset.

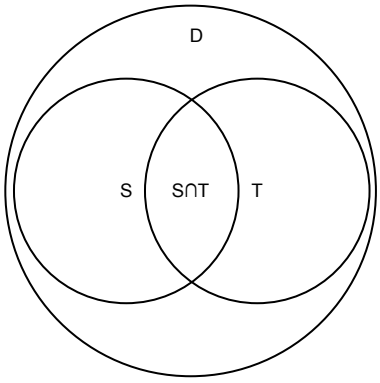


Figure 4.2: Venn diagram of the data with annotations from Songfacts $S$ and Last.fm tags $T$.

$S$ and $D$ account, for 6906 and 17795 lyrics respectively. Using the re-annotation process we now have 2054 annotated lyrics which were not previously annotated, and 842 already annotated lyrics were further re-annotated, meaning that at least 12% of annotated lyrics were suffering from partial annotation. From the 50 original categories 36 were used at least once in re-annotation.

It is this addition of annotations that will allow us to evaluate the impact of partially annotated data.

Figure 4.3 shows the category distribution after the re-annotation process, the distribution now has an entropy of 4.82 (bits) showing that the process skewed the distribution even further.

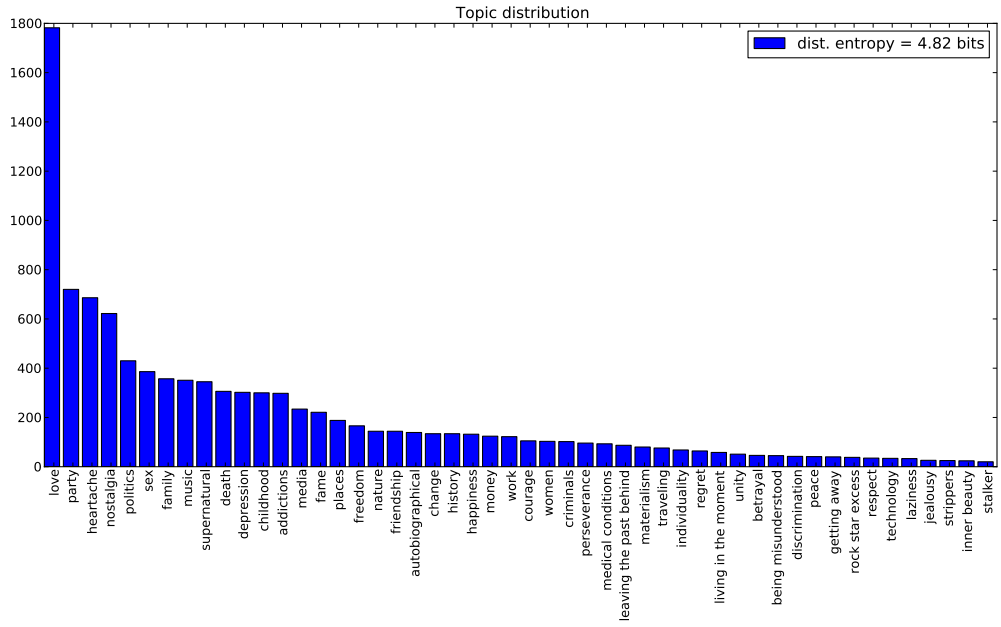


Figure 4.3: Topic distribution after re-annotation.

## 4.2 Lyrics categorization

### 4.2.1 Preprocessing

As a pre-processing step, contractions are expanded when possible and words are converted to all lowercase. We opted not to apply stemming.

### 4.2.2 Feature extraction

We will use the tf-idf weighting scheme (Section B.2.1) extracted from the songs lyrics to represent documents as vectors.

### 4.2.3 Feature selection

The only feature selection applied was the removal of punctuation and common English stop-words.

### 4.2.4 Modeling

The model chosen was a One-vs-All (Section B.2.3.2) Linear SVM (Section B.2.3.1) approach to model the categories using the extracted features.

### 4.2.5 Evaluation

To evaluate the performance of the model we will use micro averaged and non averaged Precision-Recall Curves and BEP (Section B.2.4).

## 4.3 Impact of partially annotated data

We wish to test two case scenarios. How the performance of our predictive model is affected at modeling and evaluation time, using partially annotated data.

### 4.3.1 Modeling

In this scenario we will assert the impact of fitting the model using partially annotated data. This is accomplished by using 20% of the $S \setminus S \cap T$ for test and the remaining 80% of $S$ to fit the model. In the first performance measurement the training set will contain only the annotations from Labels1, and in the second measurement the train set will have both Labels1 and Labels2 annotations.

### 4.3.2 Evaluation

In the second scenario we will use songs from $S \setminus S \cap T$ and $S \cap T$ to train and test the model. In the first performance measurement we will use just the Labels1 annotations in test set, this will serve as our baseline performance. In the second measurement we will use both Labels1 and Labels2 annotations.

# Chapter 5

# Results

## 5.1 Impact of partially annotated data in lyrics categorization

In this section we will present the results of the proposed approach, using Micro-averaged Precision-Recall curves and Interpolated Break-even Point measures for our total of 50 categories.

### 5.1.1 Modeling

In this scenario we measure the impact of partially annotated data when fitting the model. The data used to fit and test the model is the same, the only difference is in the labels given to the supervised algorithm. In Figure 5.1 the model is fitted and tested using partial annotations. In Figure 5.2 the model is fitted using the same data but with the extra labels fetched from Last.fm.

Overall the performance remained unchanged when the extra labels were added. This can be seen by the Precision-Recall Break-Even Point which remained almost identical. A detailed measurement of Precision-Recall BEP for each category is available in Figure A.1.

### 5.1.2 Evaluation

In this scenario we measure the impact of partially annotated data when testing our model. The data in both tests is the same, the only difference are the extra labels used in the test set.

Although the split is not identical to the one used in Figure 5.1, in Figure 5.3 we see a similar performance. This should be the case since we are using exactly the same data.

Unlike Figure 5.2, Figure 5.4 shows a significant increase in performance just by the addition of more labels to the initial 50 classes.

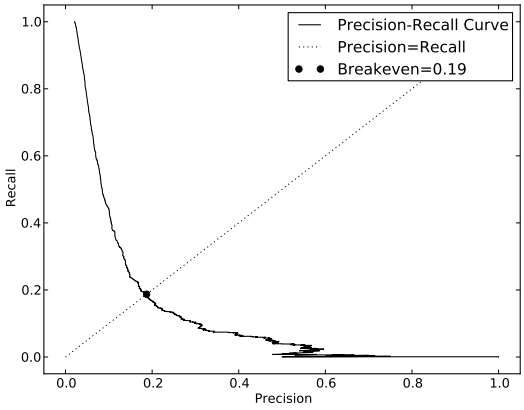


Figure 5.1: Fitting the model using partial annotations in train and test.

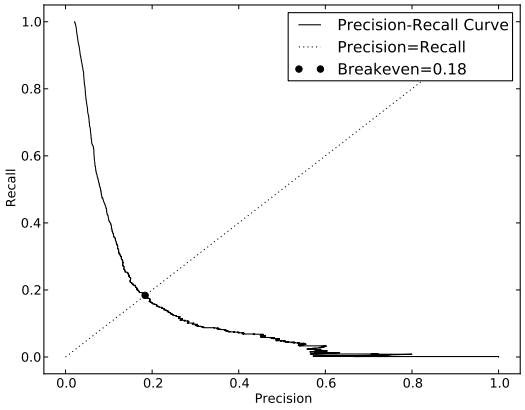


Figure 5.2: Fitting the model using added annotations and testing with partial annotations.

Even with naive approach of re-annotation the Precision-Recall BEP increases roughly 58%. A detailed measurement of Precision-Recall BEP for each category is available in Figure A.2.

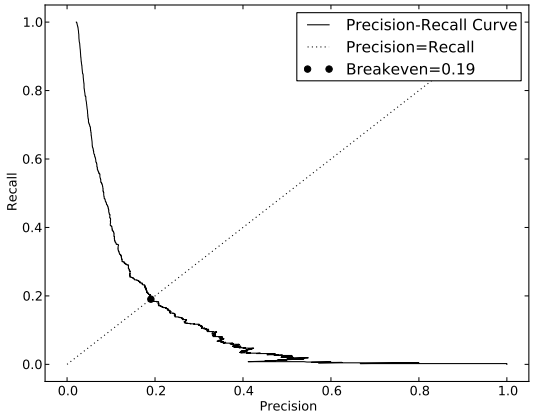


Figure 5.3: Fitting the model using partial annotations in train and test.

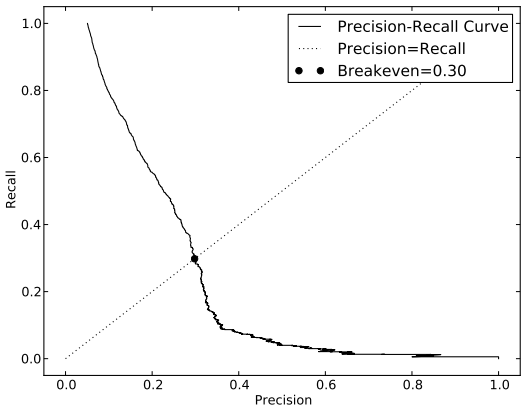


Figure 5.4: Fitting the model using partial annotations and testing using added annotations.

## 5.2 Discussion

The performance did not change much when we added more information in the training step, we argue this may be due the fact that although some observations are partially annotated, the model could identify some characteristics of each category using the combined observations.

The impact of partial annotations was more prominent when we evaluated our model. This simply means not having enough annotations grossly underestimated our model predictions.

We would also like to note the difference in performance of categorization using our specific weakly annotated lyrics dataset and previous work in categorization of news text using the Reuters dataset [1]. Having applied the same categorization methodology Figure 5.5 illustrates the Precision-Recall Curve and BEP.

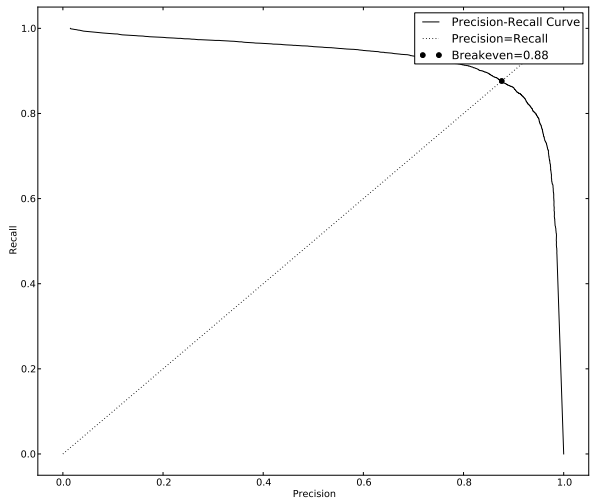


Figure 5.5: Performance of categorization in the Reuters dataset.

This difference in performance might be a reflection of the quality of annotations in each dataset or the limitations of the automatic categorization approach to poetic-text. Indeed, if we assume that poetic text is generally harder to categorize both these will occur.

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

# Chapter 6

# Conclusion

In this thesis we have measured the impact of partially annotated data when modeling and evaluating a predictive model for automatic lyrics categorization representing its semantic content. We have done this using external sources to expand annotations in a partially annotated dataset of song lyrics and using a popular text categorization supervised methodology.

We have found, in this particular setup, that partially annotated data has a much more negative impact when evaluating our predictive model, than when we model it. In other words, learning from partial annotations is more successful than evaluating. While individually some lyrics may not be annotated as belonging to a certain category those that are may be enough for the predictive model to capture the categories main characteristics. On the other hand partially annotated data will impose a bound to the performance of the model, grossly underestimating it, simply because inexistent annotations will mark correct predictions as incorrect.

We have also found that the same methodologies for text categorization and feature selection were less effective using lyrics text, then using non-poetic text. It is not obvious however if this is the consequence of the difference in the type of text or a consequence of the annotation process itself.

Although, for this specific dataset, the experiment has shown the impact of partially annotated data during the modeling phase to be negligible, we do not know how this would vary with more categories and less examples per category. In these cases we would expect the impact of partial annotations to be higher simply because we would be training our model with much more examples that were false negatives than true positives. One possible approach would be to treat this as a semi-supervised density estimation problem.

It seems certain however that the assumption of human labels as gold standard is questionable in some cases. In these cases using external sources of annotations could be beneficial in addressing partially annotations in weakly annotated data.

# Conclusion

# References


[BCD03] D. Bainbridge, S. J Cunningham, and J. S Downie. How people describe their music information needs: A grounded theory analysis of music queries. In *Proceedings of the International Symposium on Music Information Retrieval*, page 221–222, 2003.

[BNJ03] D. M Blei, A. Y Ng, and M. I Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[Hof99] Thomas Hofmann. Probabilistic latent semantic analysis. *in Proc. of Uncertainty in Artificial Intelligence, UAI'99*, pages 289—296, 1999.

[Joa98] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, page 137–142, 1998.

[Joa99] T. Joachims. Transductive inference for text classification using support vector machines. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, page 200–209, 1999.

[KKP08] F. Kleedorfer, P. Knees, and T. Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Proceedings of the International Conference on Music Information Retrieval*, page 287–292, 2008.

[LKM04] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 2, page 827–830, 2004.

[LSM10] E. Law, B. Settles, and T. Mitchell. Learning to tag from open vocabulary labels. *Machine Learning and Knowledge Discovery in Databases*, page 211–226, 2010.

[MLM07] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, page 633–640, 2007.

[MMC+05] J. P.G Mahedero, Á MartÍnez, P. Cano, M. Koppenberger, and F. Gouyon. Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, page 475–478, 2005.

[MN98] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, page 41–48, 1998.

[MNR08] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and style features for musical genre classification by song lyrics. In *In Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*, 2008.

[NMM06] K. Nigam, A. McCallum, and T. Mitchell. Semi-supervised text classification using EM. *Semi-Supervised Learning*, page 33–56, 2006.

[NMTM00] K. Nigam, A. K McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2):103–134, 2000.

[RHNM09] D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, page 248–256, 2009.

[SGO09] L. Sarmento, F. Gouyon, and E. Oliveira. Music artist tag propagation with wikipedia abstracts. In *Proceeding of the Workshop on Information Retrieval over Social Networks, European Conference on Information Retrieval (ECIR), Tolouse*, 2009.

[SM98] S. Scott and S. Matwin. Text classification using WordNet hypernyms. In *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, page 38–44, 1998.

[SM99] Sam Scott and Stan Matwin. Feature engineering for text classification. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 379–388. Morgan Kaufmann Publishers Inc., 1999. ACM ID: 657484.

[Vap00] V. N Vapnik. *The nature of statistical learning theory*. Springer Verlag, 2000.

[Ven11] Michael Sinanian Of Venturebeat. Why lyrics are the consumer web's next big thing, again. *The New York Times*, September 2011.

[WZO07] Bin Wei, Chengliang Zhang, and Mitsunori Ogihara. Keyword generation for lyrics. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR '07)*, pages 121–122, 2007.

[XLG03] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, page 267–273, 2003.

[Yan99] Y. Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1):69–90, 1999.

[YL09] Dan Yang and Won-Sook Lee. Music emotion identification from lyrics. In *Multimedia, International Symposium on*, volume 0, pages 624–629, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[YLC+08] Y. H Yang, Y. C Lin, H. T Cheng, I. B Liao, and Y. C Ho. Toward multimodal music emotion classification. *Advances in Multimedia Information Processing-PCM 2008*, page 70–79, 2008.

# REFERENCES

# Appendix A

# Results

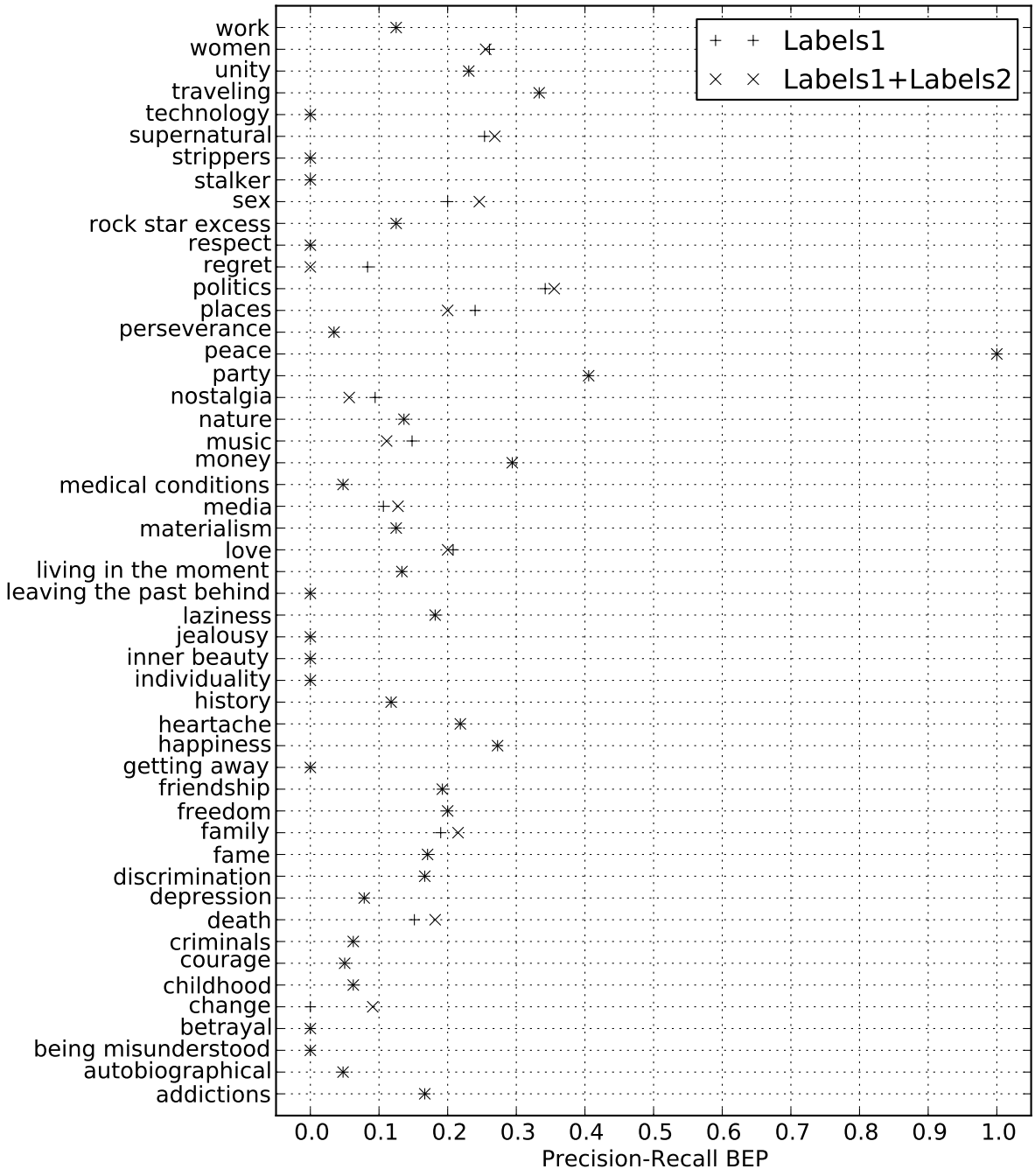


Figure A.1: Precision-Recall BEP per category when using Labels1 or using Labels1 and Labels2 in training the model.

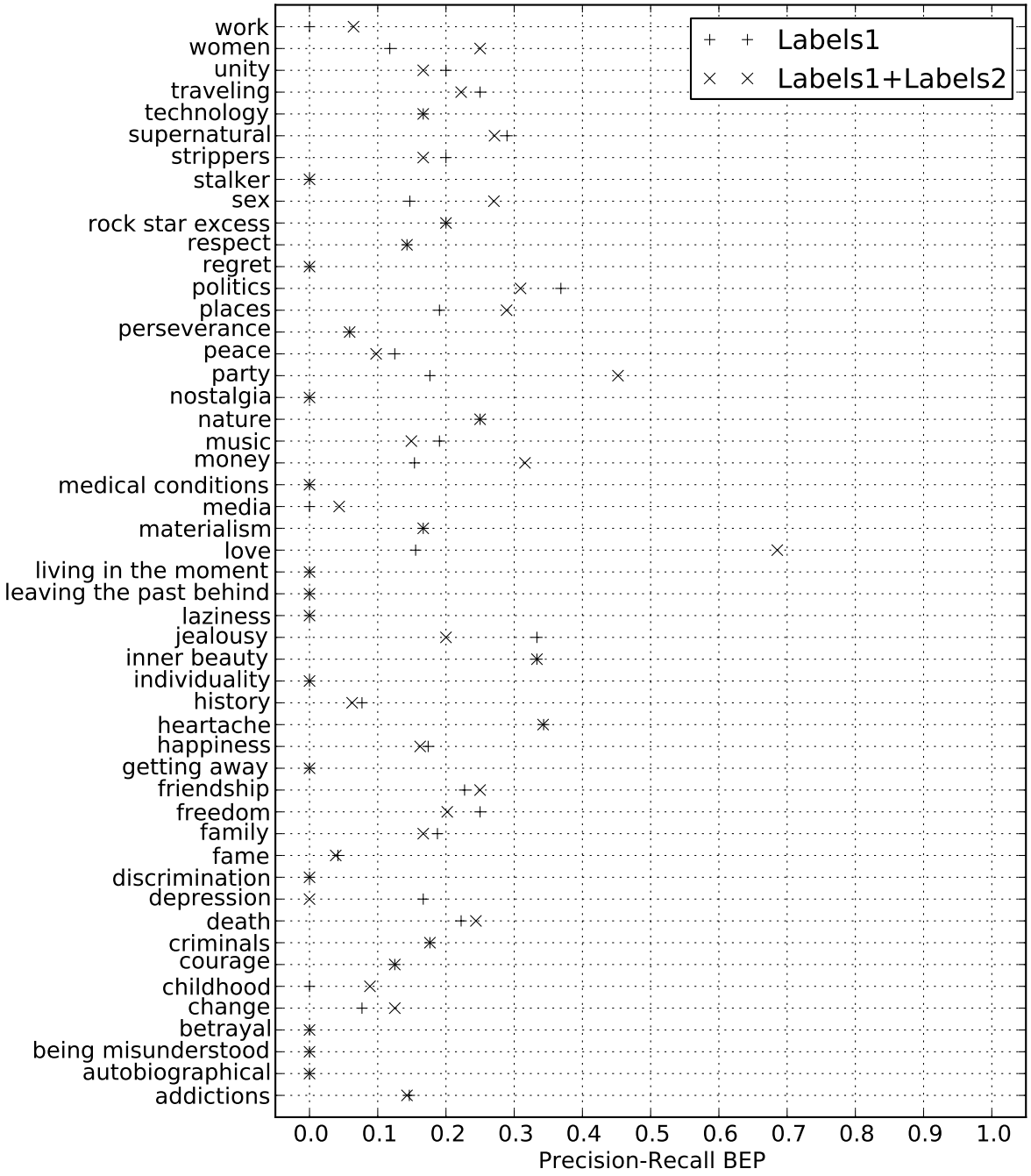


Figure A.2: Precision-Recall BEP per category when using Labels1 or using Labels1 and Labels2 in testing the model.

# Appendix B

# Techniques

## B.1 Hypothesis testing

Statistical hypothesis testing is a method of using data to evaluate a given hypothesis. We use the data as a sample of a population we wish to test our hypothesis with, and the result of such a test is said to be significant, or unlikely to have occurred by chance, by using a threshold.

The usual procedure is to define two hypothesis, the null hypothesis $H_0$ which we want to test if it is true, and the alternative hypothesis $H_1$. Only by finding a result which is very unlikely to have happened by chance can we reject the null hypothesis.

### B.1.1 Pearson's chi-squared test

The Pearson's chi-squared test ($\chi^2$), is a statistical procedure which tests if a sampled distribution is consistent with a theoretical one, in which all events must be mutually exclusive (probabilities adding to one).

#### B.1.1.1 Fit of distribution

For example if our null hypothesis is that of a uniform distribution we would define the expected frequency of event i as:

$$E_i = \frac{N}{n} \tag{B.1}$$

Where $N$ is our sample size and $n$ is the number of events in the distribution. This means each event should have the same frequency of counts, since our theoretical distribution is uniform.

The test-statistic is defined as

$$X^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{B.2}$$

Where $O_i$ are the observed frequencies for each event.

Using our test statistic we can calculate the p-value of the associated $\chi^2$ distribution. The degrees of freedom k of the distribution are the difference between the number of events $n$ and the number of parameters $p$ used in the theoretical distribution plus one ($k = n - (p+1)$).

The p-value tells us how likely we are to have obtained a test-statistic as extreme as the observed one, assuming that our null hypothesis is true. Usually a p-value of 0.05 or less shows evidence that our null hypothesis is false.

#### B.1.1.2 Test of independence

We may also wish to test the independence between two discrete random variables.

Consider the random variables $T$ and $C$ which can take two possible values, 1 and 0.

By building a two-by-two table with the observed counts of the events we can calculate both the joint and marginal distributions for $T$ and $C$, this will be useful since our null hypothesis assumes that $p(T,C) = p(T) \times p(C)$ (the variables are independent).

By using the joint distribution as the observed distribution and the product of both marginals as the expected distribution (Equation B.3) we can use the Pearson's chi-squared test to find statistical evidence to reject $H_0$ (Equation B.4).

$$E_{tc} = N \times p(t) \times p(c) = \frac{\sum_i O_{ti} \times \sum_j E_{jc}}{N} \tag{B.3}$$

$$X^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{B.4}$$

By using the test-statistic with a $\chi_1^2$ distribution, were 1 is the degree of freedom, we find the p-value. If the p-value for a pair term-class is less than 0.05 we can say there is evidence that the two are not independent.

## B.2 Automatic Text Categorization

Automatic text categorization is the automatic categorization of documents of text into a pre-defined set of categories.

In this section we will talk about some of the steps involved in creating and evaluating a predictive model for this task.

For notation purposes we will define our set of $m$ documents as matrix $X^{m,v}$, where each row represents a document as a column vector of $v$ dimensions.

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \dots \\ (x^{(m)})^T \end{bmatrix}$$

Matrix $Y^{m,c}$ is a binary valued matrix that contains the correspondence between documents and categories such that $Y_{i,j} = 1$ if document $i$ belongs to category $j$.

$$Y = \begin{bmatrix} (y^{(1)})^T \\ (y^{(2)})^T \\ \dots \\ (y^{(m)})^T \end{bmatrix}$$

When we want to refer to the $i^{th}$ document we use the notation $x^{(i)}$ or $y^{(i)}$, and when we want to refer to the $j^{th}$ attribute or dimension we will use $x_j$ or $y_j$.

When we wish to refer to a specific word in a text, we will use word. When referring to a word which occurs at least once in any document we will use term. So that the set of all terms forms our vocabulary.

### B.2.1 Feature extraction

The first step in text categorization is the extraction of meaningful features that represent a document of text. What a meaningful feature is depends on how we will model the data, in general and in this case, meaningful features are the terms in our vocabulary.

For instance $x_j^{(i)}$ contains the value of feature (dimension) $j$ of document $i$. All feature values of all documents will form matrix $X$. Typical values used in this feature space are raw and weighted measures.

An example of raw measure could be the number of times term $t$ appears in document $d$ (Equation B.5).

$$\text{tf}_{t,d} = |\{i : i \in d \ \wedge t = i\}| \tag{B.5}$$

Weighted measures not only take into account how much a term occurs but also the terms overall importance. Terms which appear in few documents are considered more important (rarer) than terms which appear in a high number of documents. A specific example is called the tf-idf (Equation B.6), which is the product of the normalized term frequency and its inverse document frequency logarithm.

$$\text{tf-idf}_{t,d} = \frac{\text{tf}_{t,d}}{\sum_k \text{tf}_{k,d}} \times \log \frac{m}{|\{i : t \in x^{(i)}\}|} \tag{B.6}$$

### B.2.2 Feature selection

Feature selection is a procedure to select a subset of the best or most informative features.

Remotion of terms with extremely high occurrence (also known as stop-words) or low occurrence (terms that only appear once) are an easy and commonly used method in text categorization. Other approaches include using Pearson's Chi Square test to select the words with the lowest p-values.

Some authors however do not advise aggressive feature selection. Even using only "poor" features to predict categories outperforms random guessing [Joa98].

### B.2.3 Modeling

Generally speaking, modeling is the search for an hypothesis in the hypothesis space. An hypothesis is simply a function which outputs predictions using as input new data. Although this is not always the case, hypothesis searching could be simply finding a model parameters which minimize a given loss function.

Generally speaking, approaches like supervised, semi-supervised or unsupervised are simply distinctions made on how data is used in searching the hypothesis space. Supervised approaches use data which contains the correct predictions, semi-supervised only has access to some data predictions and unsupervised has no predictions available.

Supervised [MN98, Joa98], semi-supervised [Joa99, NMM06] and unsupervised [BNJ03, MLM07, RHNM09] approaches have all been used in, but not restricted to, text categorization.

#### B.2.3.1 SVM

Support Vector Machine (SVM) [CV95, Vap00] is a supervised learning algorithm.

Originally the algorithm linearly separated data belonging to two possible classes using a maximal margin hyperplane. As the margin between data and the hyperplane increases, so does the confidence of our worst prediction.

Because not all data is linearly separable, the algorithm was latter extended with the use of soft-margins which account for misclassifications.

SVM is called an optimal margin classifier because of its geometric interpretation (Figure B.1) and is solved as an optimization problem, in which our objective function minimizes a trade-off between the complexity of the model (the normal vector of the hyperplane) and the number of misclassifications.

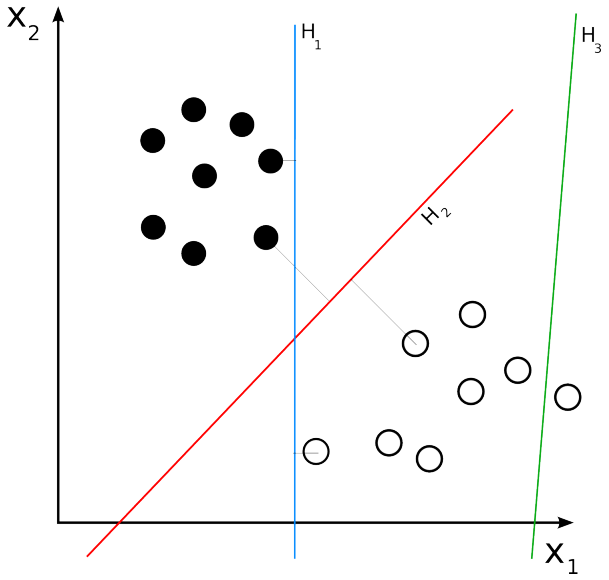


Figure B.1: Hyperplanes separating synthetic positive and negative instances. Hyperplane that maximizes the distance of the support vectors is chosen.

Having solved this optimization problem and found the hyperplane, predicting the class for new data depends only on inner products with the vectors that reside closest to the hyperplane, these are called the support vectors.

Although our data may not be linearly separable there may exist a space in which they are (Figure B.2), this transformation between spaces is represented by $\phi$. When finding the hyperplane and making predictions instead of calculating inner products $\langle z,x\rangle$ we just calculate the inner product of the points in the new space $\langle \phi(z), \phi(x)\rangle$.

We define the kernel function as

$$K(z,x) = \phi(z)^T \phi(x)$$

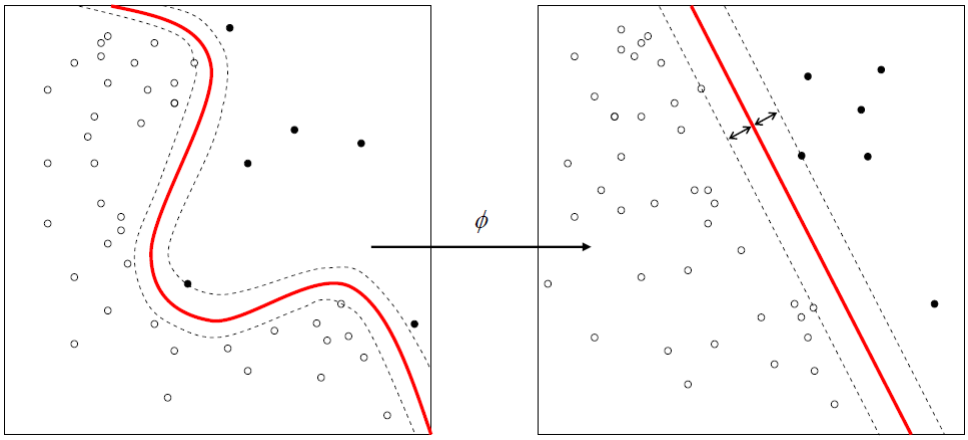


Figure B.2: Transformation $\phi$ of original space into a new space H, where all synthetic examples are linearly separable.

Some examples of kernels used are:

**Linear Kernel** $K(z,x) = \langle z,x \rangle$

**Polynomial Kernel** $K(z,x) = (\langle z,x \rangle + r)^d$. Where $d$ is the degree of the polynomial.

**Radial Basis Function Kernel** $K(z,x) = exp(-\gamma|z-x|^2)$ for $\gamma > 0$. When $\gamma = \frac{1}{2\sigma^2}$ this corresponds to the Gaussian function.

#### B.2.3.2 Multiclass and Multilabel

Originally SVMs are used for binary classification, but it is also possible to use it in a multiclass scenario like the one in text categorization. One possible approach is to create a classifier for each category and when predicting new data we assign the category of the classifier with the highest confidence. This is called the one-versus-all (OvA) approach.

Attributing multiple categories to a document is done by using a minimum confidence threshold and instead of assigning the category of the classifier with the highest confidence we assign the categories of the classifiers which have a higher confidence then the threshold.

### B.2.4 Evaluation measures

#### B.2.4.1 Measures

**True Positives** (TP) are the number of items classified as True that are indeed True. An example would be a spam email classified as spam.

**False Positives** (FP) are the number of items classified as True that are not True. For instance a non spam email being classified as spam.

**True Negatives** (TN) are the number of items classified as False that are in reality False. For instance a non spam email not being classified as spam.

**False Negatives** (FN) are the number of items classified as False that are True. A spam email being not being classified as spam.

Although this assumes a binary classification task, the generalization of this to multi-class classification considers any given class as True and all the others as False. For visualization purposes a confusion matrix represents this information in a table like fashion as Table B.1 illustrates.

| | | Predicted | | |
|---|---|---|---|---|
| | | A | B | C |
| Actual | A | $TP_A$ | $E_{A,B}$ | $E_{A,C}$ |
| | B | $E_{B,A}$ | $TP_B$ | $E_{B,C}$ |
| | C | $E_{C,A}$ | $E_{C,B}$ | $TP_C$ |

Table B.1: Multiclass confusion matrix

By looking at the confusion matrix in B.1 we can calculate in each class the same measures as for the binary task. For instance class A is characterized by:

**True Positives A** $TP_A$

**False Positives A** $FP_A = E_{B,A} + E_{C,A}$

**True Negatives A** $TN_A = TP_B + E_{B,C} + E_{C,B} + TP_C$

**False Negatives A** $FN_A = E_{A,B} + E_{A,C}$

The previous measures are then used to calculate some useful metrics that summarize the performance of the classifier.

**Accuracy** represents the probability of the classifier correctly classifying a given element. This measure is not often used because the true negatives high count often bias the result too much.

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \tag{B.7}$$

**Precision** is the probability of an item classified as relevant actually being relevant.

$$\text{Precision} = \frac{tp}{tp + fp} \tag{B.8}$$

**Recall** (also called True positive rate or TPR) represents how likely any relevant item is to be classified as relevant.

$$\text{Recall} = \frac{tp}{p} = \frac{tp}{tp + fn} \tag{B.9}$$

**False positive rate** (FPR or type I error rate) represents how likely any non relevant item is to be classified as non relevant.

$$\text{FPR} = \frac{fp}{n} = \frac{fp}{fp + tn} \tag{B.10}$$

**F-measure** tries to represent the precision and accuracy in one value using $\alpha \in [0,1]$. When $\alpha < 0.5$ we are giving more importance to recall, when $\alpha > 0.5$ we are weighting more precision. The function $\beta$ can also be seen as the number of times recall is more important than precision.

$$\beta^2 = \frac{1}{\alpha} - 1 \tag{B.11}$$

$$\text{F-measure} = \frac{1}{1+\beta^2} \times \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{B.12}$$

$\mathbf{F}_1$ is the F-measure when $\alpha = 0.5$, which corresponds to the precision and recall harmonic mean.

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{B.13}$$

This measure is maximized when both values are equal, otherwise the smaller value dominates the measure. [Yan99]

**Precision-Recall Curve** When using a threshold to assign categories to a document, by varying its value we can trade-off precision for recall and vice-versa.

By starting with a very high threshold we only assign categories to a document if we see a very confident classifier, so precision should be high while recall low. By slowly relaxing this threshold precision will drop, but recall will go up, and when the threshold reaches very low values we should see a very low precision and a perfect recall.

The Precision-Recall Curve shows this trade-off using different thresholds B.3.

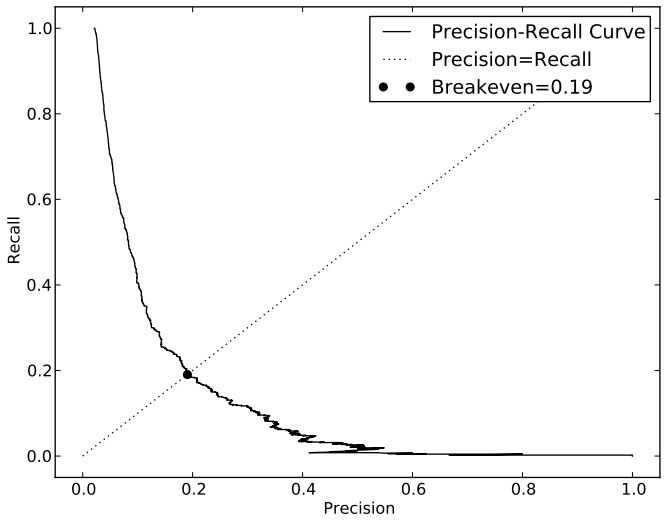


Figure B.3: Precision-Recall Curve represents the trade-off of Precision and Recall. Example from chapter 5.

**Precision-Recall Break-even point (BEP)** For a certain threshold the Precision and Recall are even, we call this the Precision-Recall Break-even point. This is illustrated in Figure B.3. If these measures cannot be made exactly equal, the Interpolated BEP is used which is simply the arithmetic mean of the closes values of Precision and Recall.

Recalling that the BEP is just the $F_1$ measure when precision and recall are even and at this point the $F_1$ reaches its maximum then BEP $\geq F_1$. This is intuitive since the arithmetic mean is always greater or equal than the harmonic mean.

#### B.2.4.2 Averaging performance

All these measures can be applied to evaluate a specific category or can be averaged over all categories to provide an overall measure.

There are two main approaches to average these measures, macro and micro averaging.

Macro averaging simply averages the measure over all the categories while micro averaging builds a confusion matrix summing all categories $TP$, $TN$, $FP$ and $FN$ and calculates the measure based on those values.

Equation B.14 and B.15 show how we would micro and macro average the precision measure over $|C|$ multiple categories.

$$P_{\text{micro}} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i} \tag{B.14}$$

$$P_{\text{macro}} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FP_i} \tag{B.15}$$