Combining labelled and unlabelled data: a case study on Fisher kernels and transductive inference for biological entity recognition

Cyril Goutte, Hervé Déjean, Eric Gaussier,
Nicola Cancedda and **Jean-Michel Renders**
Xerox Research Center Europe
6, chemin de Maupertuis
38240 Meylan, France

Abstract

We address the problem of using partially labelled data, eg large collections were only little data is annotated, for extracting biological entities. Our approach relies on a combination of probabilistic models, which we use to model the generation of entities and their context, and kernel machines, which implement powerful categorisers based on a similarity measure and some labelled data. This combination takes the form of the so-called *Fisher kernels* which implement a similarity based on an underlying probabilistic model. Such kernels are compared with transductive inference, an alternative approach to combining labelled and unlabelled data, again coupled with Support Vector Machines. Experiments are performed on a database of abstracts extracted from Medline.

1 Introduction

The availability of electronic databases of rapidly increasing sizes has encouraged the development of methods that can tap into these databases to automatically generate knowledge, for example by retrieving relevant information or extracting entities and their relationships. Machine learning seems especially relevant in this context, because it helps performing these tasks with a minimum of user interaction.

A number of problems like entity extraction or filtering can be mapped to supervised techniques like categorisation. In addition, modern supervised classification methods like Support Vector Machines have proven to be efficient and versatile. They do, however, rely on the availability of labelled data, where labels indicate eg whether a document is relevant or whether a candidate expression is an interesting entity. This causes two important problems that motivate our work: 1) annotating data is often a difficult and costly task involving a lot of human work[1], such that large collections of labelled data are difficult to obtain, and 2) inter-annotator agreement tends to be low in eg genomics collections (Krauthammer et al., 2000), thus calling for methods that are able to deal with noise and incomplete data.

On the other hand, unsupervised techniques do not require labelled data and can thus be applied regardless of the annotation problems. Unsupervised learning, however, tend to be less data-efficient than its supervised counterpart, requiring many more examples to discover significant features in the data, and is incapable of solving the same kinds of problems. For example, an efficient clustering technique may be able to distribute documents in a number of well-defined clusters. However, it will be unable to decide which clusters are relevant without a minimum of supervision.

This motivates our study of techniques that rely on a combination of supervised and unsupervised learning, in order to leverage the availability of large collections of unlabelled data and use a limited amount of labelled documents.

The focus of this study is on a particular application to the genomics literature. In genomics, a vast amount of knowledge still resides in large collections of scientific papers such as Medline, and several approaches have been proposed to extract, (semi-)automatically, information from such papers. These approaches range from purely statistical ones to symbolic ones relying on linguistic and knowledge processing tools (Ohta et al., 1997; Thomas et al., 2000; Proux et al., 2000, for example). Furthermore, due to the nature of the problem at hand, meth-

[1] If automatic annotation was available, we would basically have solved our Machine Learning problem

ods derived from machine learning are called for, (Craven and Kumlien, 1999), whether supervised, unsupervised or relying on a combination of both.

Let us insist on the fact that our work is primarily concerned with combining labelled and unlabelled data, and entity extraction is used as an application in this context. As a consequence, it is not our purpose at this point to compare our experimental results to those obtained by specific machine learning techniques applied to entity extraction (Califf, 1999). Although we certainly hope that our work can be useful for entity extraction, we rather think of it as a methodological study which can hopefully be applied to different applications where unlabelled data may be used to improve the results of supervised learning algorithms. In addition, performing a fair comparison of our work on standard information extraction benchmarks is not straightforward: either we would need to obtain a large amount of unlabelled data that is comparable to the benchmark, or we would need to "un-label" a portion of the data. In both cases, comparing to existing results is difficult as the amount of information used is different.

2 Classification for entity extraction

We formulate the following (binary) classification problem: given an input space $\mathcal{X}$, and from a dataset of N input-output pairs $(x_k, y_k) \in \mathcal{X} \times \{-1; +1\}$, we want to learn a classifier $h : \mathcal{X} \to \{-1; +1\}$ so as to maximise the probability $P(h(x) = y)$ over the fixed but unknown joint input-output distribution of (x, y) pairs. In this setting, binary classification is essentially a supervised learning problem.

In order to map this to the biological entity recognition problem, we consider for each candidate term, the following binary decision problem: is the candidate a biological entity[2] $(y = 1)$ or not $(y = -1)$. The input space is a high dimensional feature space containing lexical, morpho-syntactic and contextual features.

In order to assess the validity of combining labelled and unlabelled data for the particular task of biological entity extraction, we use the following tools. First we rely on Suport Vector Machines together with transductive inference (Vapnik, 1998; Joachims, 1999), a training technique that takes both labelled and unlabelled data into account. Secondly, we develop a Fisher kernel (Jaakkola and Haussler, 1999), which derives the similarity from an underlying (unsupervised) model of the data, used as a similarity measure (aka kernel) within SVMs. The learning process involves the following steps:

- Transductive inference: learn a SVM classifier $h(x)$ using the combined (labelled and unlabelled) dataset, using traditional kernels.
- Fisher kernels:
  1. Learn a probabilistic model of the data $P(x|\theta)$ using combined unlabelled and labelled data;
  2. Derive the Fisher kernel $K(x, z)$ expressing the similarity in $\mathcal{X}$-space;
  3. Learn a SVM classifier $h(x)$ using this Fisher kernel and inductive inference.

3 Probabilistic models for co-occurence data

In (Gaussier et al., 2002) we presented a general hierarchical probabilistic model which generalises several established models like Naïve Bayes (Yang and Liu, 1999), probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) or hierarchical mixtures (Toutanova et al., 2001). In this model, data result from the observation of co-occuring objects. For example, a document collection is expressed as co-occurences between documents and words; in entity extraction, co-occuring objects may be potential entities and their context, for example. For co-occuring objects i and j, the model is expressed as follows:

$$P(i, j) = \sum_{\alpha} P(\alpha)\, P(i|\alpha) \sum_{\nu} P(\nu|\alpha)\, P(j|\nu) \tag{1}$$

where α are latent classes for co-occurrences (i, j) and ν are latent nodes in a hierarchy generating objects j. In the case where no hierarchy is needed (ie $P(\nu|\alpha) = \delta(\nu = \alpha)$), the model reduces to PLSA:

$$P(i, j) = \sum_{\alpha} P(\alpha)\, P(i|\alpha)\, P(j|\alpha) \tag{2}$$

[2] In our case, biological entities are proteins, genes and RNA, cf. section 6.

where α are now latent concepts over both i and j. Parameters of the model (class probabilities $P(\alpha)$ and class-conditional $P(i|\alpha)$ and $P(j|\alpha)$) are learned using a deterministic annealing version of the expectation-maximisation (EM) algorithm (Hofmann, 1999; Gaussier et al., 2002).

4 Fisher kernels

Probabilistic generative models like PLSA and hierarchical extensions (Gaussier et al., 2002) provide a natural way to model the generation of the data, and allow the use of well-founded statistical tools to learn and use the model.

In addition, they may be used to derive a model-based measure of similarity between examples, using the so-called Fisher kernels proposed by Jaakkola and Haussler (1999). The idea behind this kernel is that using the structure implied by the generative model will give a more relevant similarity estimate, and allow kernel methods like the support vector machines or nearest neighbours to leverage the probabilistic model and yield improved performance (Hofmann, 2000).

The Fisher kernel is obtained using the log-likelihood of the model and the Fisher information matrix. Let us consider our collection of documents $\{x_k\}_{k=1...N}$, and denote by $\ell(x) = \log P(x|\theta)$ the log-likelihood of the model for data x. The expression of the Fisher kernel (Jaakkola and Haussler, 1999) is then:

$$K(x_1, x_2) = \nabla\ell(x_1)^\top \, \mathbf{I_F}^{-1} \, \nabla\ell(x_2) \qquad (3)$$

The Fisher information matrix $\mathbf{I_F}$ can be seen as a way to keep the kernel expression independent of parameterisation and is defined as $\mathbf{I_F} = \mathsf{E}\left(\nabla\ell(x)\,\nabla\ell(x)^\top\right)$, where the gradient is w.r.t. θ and the expectation is taken over $P(x|\theta)$. With a suitable parameterization, the information matrix $\mathbf{I}$ is usually approximated by the identity matrix (Hofmann, 2000), leading to the simpler kernel expression: $K(x_1, x_2) = \nabla\ell(x_1)^\top \nabla\ell(x_2)$.

Depending on the model, the various log-likelihoods and their derivatives will yield different Fisher kernel expressions. For PLSA (2), the parameters are $\theta = [P(\alpha), P(i|\alpha), P(j|\alpha)]$. From the derivatives of the likelihood $\ell(x) = \sum_{(i,j)\in x} \log P(i,j)$, we derive the following similarity (Hofmann, 2000):

$$K(x_1, x_2) = \sum_\alpha \frac{P(\alpha|d_i)\,P(\alpha|d_j)}{P(\alpha)} \qquad (4)$$
$$+ \quad \sum_w \widehat{P}_{wd_i}\,\widehat{P}_{wd_j} \sum_\alpha \frac{P(\alpha|d_i, w)\,P(\alpha|d_j, w)}{P(w|\alpha)}$$

with $\widehat{P}_{wd_i}$, $\widehat{P}_{wd_j}$ the empirical word distributions in documents d_i, d_j.

5 Transductive inference

In standard, inductive SVM inference, the annotated data is used to infer a model, which is then applied to unannotated test data. The inference consists in a trade-off between the size of the margin (linked to generalisation abilities) and the number of training errors. Transductive inference (Gammerman et al., 1998; Joachims, 1999) aims at maximising the margin between positives and negatives, while minimising not only the actual number of incorrect predictions on labelled examples, but also the expected number of incorrect predictions on the set of unannotated examples.

This is done by including the unknown labels as extra variables in the original optimisation problem. In the linearly separable case, the new optimisation problem amounts now to find a labelling of the unannotated examples and a hyperplane which separates all examples (annotated and unannotated) with maximum margin. In the non-separable case, slack variables are also associated to unannotated examples and the optimisation problem is now to find a labelling and a hyperplane which optimally solves the trade-off between maximising the margin and minimising the number of misclassified examples (annotated and unannotated).

With the introduction of unknown labels as supplementary optimisation variables, the constraints of the quadratic optimisation problem are now nonlinear, which makes solving more difficult. However, approximated iterative algorithms exist which can efficiently train Transductive SVMs. They are based on the principle of gradually improving the solution by switching the labels of unnannotated examples which are misclassified at the current iteration, starting from an initial labelling given by the standard (inductive) SVM.

WUp	Is the word capitalized?
WAllUp	Is the word alls capitals?
WNum	Does the word contain digits?

Table 1: Spelling features

6 Experiments

For our experiments, we used 184 abstracts from the Medline site. In these articles, genes, proteins and RNAs were manually annotated by a biologist as part of the BioMIRE project. These articles contain 1405 occurrences of gene names, 792 of protein names and 81 of RNA names. All these entities are considered relevant biological entities. We focus here on the task of identifying names corresponding to such entities in running texts, without differentiating genes from proteins or RNAs. Once candidates for biological entity names have been identified, this task amounts to a binary categorisation, relevant candidates corresponding to biological entity names. We divided these abstracts in a training and development set (122 abstracts), and a test set (62 abstracts). We then retained different portions of the training labels, to be used as labelled data, whereas the rest of the data is considered unlabelled.

6.1 Definition of features

First of all, the abstracts are tokenised, tagged and lemmatized. Candidates for biological entity names are then selected on the basis of the following heuristics: *a token is considered a candidate if it appears in one of the biological lexicons we have at our diposal, or if it does not belong to our general English lexicon.* This simple heuristics allows us to retain 93% (1521 out of 1642) of biological names in the training set (90% in the test set), while considering only 21% of all possible candidates (5845 out of 27350 tokens). It thus provides a good pre-filter which significantly improves the performance, in terms of speed, of our system. The biological lexicons we use were provided by the BioMIRE project, and were derived from the resources available at: `http://iubio.bio.indiana.edu/`.

For each candidate, three types of features were considered. We first retained the part-of-speech and some spelling information (table 1). These features were chosen based on the inspection of gene and protein names in our lexicons.

LexPROTEIN	Protein lexicon
LexGENE	Gene lexicon
LexSPECIES	Biological species lexicon
LEXENGLISH	General English lexicon

Table 2: Features provided by lexicons.

The second type of features relates to the presence of the candidate in our lexical resources[3] (table 2). Lastly, the third type of features describes contextual information. The context we consider contains the four preceding and the four following words. However, we did not take into account the position of the words in the context, but only their presence in the right or left context, and in addition we replaced, whenever possible, each word by a feature indicating (a) whether the word was part of the gene lexicon, (b) if not whether it was part of the protein lexicon, (c) if not whether it was part of the species lexicon, (d) and if not, whenever the candidate was neither a noun, an adjective nor a verb, we replaced it by its part-of-speech.

For example, the word *hairless* is associated with the features given in Table 3, when encountered in the following sentence: *Inhibition of the DNA-binding activity of Drosophila suppressor of* hairless *and of its human homolog, KBF2/RBP-J kappa, by direct protein–protein interaction with Drosophila hairless.* The word *hairless* appears in the gene lexicon and is wrongly recognized as an adjective by our tagger.[4] The word *human*, the fourth word of the right context of *hairless*, belongs to the species lexicon, ans is thus replaced by the feature *RC_SPECIES*. Neither *Drosophila* nor *suppressor* belong to the specialized lexicons we use, and, since they are both tagged as nouns, they are left unchanged. Prepositions and conjunctions are replaced by their part-of-speech, and prefixes *LC_* and *RC_* indicate whether they were found in left or right context. Note that since two prepositions appear in the left context of *hairless*, the value of the *LC_PREP* feature is 2.

Altogether, this amounts to a total of 3690 possible features in the input space $\mathcal{X}$.

[3] Using these lexicons alone, the same task with the same test data, yields: precision = 22%, recall = 76%.

[4] Note that no adaptaion work has been conducted on our tagger, which explains this error.

Feature	Value
LexGENE	1
ADJ	1
LC_drosophila	1
LC_suppressor	1
LC_PREP	2
RC_CONJ	1
RC_SPECIES	1
RC_PRON	1
RC_PREP	1

Table 3: Features of *hairless* in "*...of Drosophila suppressor of hairless and of its human...*".

6.2 Results

In our experiments, we have used the following methods:

- SVM trained with inductive inference, and using a linear kernel, a polynomial kernel of degree $d = 2$ and the so-called "radial basis function" kernel (Schölkopf and Smola, 2002).
- SVM trained with transductive inference, and using a linear kernel or a polynomial kernel of degree $d = 2$.
- SVM trained with inductive inference using Fisher kernels estimated from the whole training data (without using labels), with different number of classes c in the PLSA model (4).

The proportion of labelled data is indicated in the tables of results. For SVM with inductive inference, only the labelled portion is used. For transductive SVM (TSVM), the remaining, unlabelled portion is used (without the labels). For the Fisher kernels (FK), an unsupervised model is estimated on the full dataset using PLSA, and a SVM is trained with inductive inference on the labelled data only, using the Fisher kernel as similarity measure.

6.3 Transductive inference

Table 4 gives interesting insight into the effect of transductive inference. As expected, in the limit where little unannotated data is used (100% in the table), there is little to gain from using transductive inference. Accordingly, performance is roughly equivalent[5]for SVM and TSVM, with a slight advantage for RBF kernel trained with inductive inference. Interestingly, in the other limit, ie when very little annotated data is used, transductive inference does not seem to yield a marked improvement over inductive learning. This finding seems somehow at odds with the results reported by Joachims (1999) on a different task (text categorisation). We interpret this result as a side-effect of the search strategy, where one tries to optimise both the size of the margin and the labelling of the unannotated examples. In practice, an exact optimisation over this labelling is impractical, and when a large amount of unlabelled data is used, there is a risk that the approximate, sub-optimal search strategy described by Joachims (1999) may fail to yield a solution that is markedly better that the result of inductive inference.

% annotated:	1.5%	6%	24%	100%
SVM (lin)	41.22	45.34	49.67	62.97
SVM (d=2)	40.97	46.78	52.12	62.69
SVM (rbf)	42.51	49.53	51.11	63.96
TSVM (lin)	38.63	51.64	61.84	62.91
TSVM (d=2)	43.88	52.38	55.36	62.72

Table 4: F_1 scores(in %) using different proportions of annotated data for the following models: SVM with inductive inference (SVM) and linear (lin) kernel, second degree polynomial kernel (d=2), and RBF kernel (rbf); SVM with transductive inference (TSVM) and linear (lin) kernel or second degree polynomial (d=2) kernel.

For the two intermediate situation, however, transductive inference seems to provide a sizeable performance improvement. Using only 24% of annotated data, transductive learning is able to train a linear kernel SVM that yields approximately the same performance as inductive inference on the full annotated dataset. This means that we get comparable performance using only what corresponds to about 30 abstracts, compared to the 122 of the full training set.

6.4 Fisher kernels

The situation is somewhat different for SVM trained with inductive inference, but using

[5] Performance is not strictly equivalent because SVM and TSVM use the data differently when optimising the trade-off parameter C over a validation set.

% annotated:	1.5%	6%	24%	100%
SVM (lin)	41.22	45.34	49.67	62.97
SVM (d=2)	40.97	46.78	52.12	62.69
lin+FK8	46.08	42.83	54.59	63.92
lin+FK16	44.43	40.92	55.70	63.76
lin+combi	46.38	38.10	52.74	63.08

Table 5: F_1 scores(in %) using different proportions of annotated data for the following models: standard SVM with linear (lin) and second degree polynomial kernel (d=2); Combination of linear kernel and Fisher kernel obtained from a PLSA with 4 classes (lin+FK4) or 8 classes (lin+FK8), and combination of linear and all Fisher kernels obtained from PLSA using 4, 8, 12 and 16 classes (lin+combi).

Fisher kernels obtained from a model of the entire (non-annotated) dataset. As the use of Fisher kernels alone was unable to consistently achieve acceptable results, the similarity we used is a combination of the standard linear kernel and the Fisher kernel (a similar solution was advocate by Hofmann (2000)). Table 5 summarises the results obtained using several types of Fisher kernels, depending on how many classes were used in PLSA. FK8 (resp. FK16) indicates the model using 8 (resp. 16) classes, while combi is a combination of the Fisher kernels obtained using 4, 8, 12 and 16 classes.

The effect of Fisher kernels is not as clear-cut as that of transductive inference. For fully annotated data, we obtain results that are similar to the standard kernels, although often better than the linear kernel. Results obtained using 1.5% and 6% annotated data seem somewhat inconsistent, whith a large improvement for 1.5%, but a marked degradation for 6%, suggesting that in that case, adding labels actually hurts performance. We conjecture that this may be an artifact of the specific annotated set we selected. For 24% annotated data, the Fisher kernel provides results that are inbetween inductive and transductive inference using standard kernels.

7 Discussion

The results of our experiments are encouraging in that they suggest that both transductive inference and the use of Fisher kernels are potentially effective way of taking unannotated data into account to improve performance.

These experimental results suggest the following remark. Note that Fisher kernels can be implemented by a simple scalar product (linear kernel) between Fisher scores $\nabla\ell(x)$ (equation 3). The question arises naturally as to whether using non-linear kernels may improve results. One one hand, Fisher kernels are derived from information-geometric arguments (Jaakkola and Haussler, 1999) which require that the kernel reduces to an inner-product of Fisher scores. On the other hand, polynomial and RBF kernels often display better performance than a simple dot-product. In order to test this, we have performed experiments using the same features as in section 6.4, but with a second degree polynomial kernel. Overall, results are consistently worse than before, which suggest that the expression of the Fisher kernel as the inner product of Fisher scores is theoretically well-founded and empirically justified.

Among possible future work, let us mention the following technical points:

1. Optimising the weight of the contributions of the linear kernel and Fisher kernel, eg as $K(x, y) = \alpha \langle x, y\rangle + (1-\alpha)FK(x, y)$, $\alpha \in [0, 1]$.
2. Understanding why the Fisher kernel alone (ie without interpolation with the linear kernel) is unable to provide a performance boost, despite attractive theoretical properties.

In addition, the performance improvement obtained by both transductive inference and Fisher kernels suggest to use both in cunjunction. To our knowledge, the question of whether this would allow to "bootstrap" the unlabelled data by using them twice (once for estimating the kernel, once in transductive learning) is still an open research question.

Finally, regarding the application that we have targeted, namely entity recognition, the use of additional unlabelled data may help us to overcome the current performance limit on our database. None of the additional experiments conducted internally using probabilisitc models and symbolic, rule-based methods have been able to yield F_1 scores higher than 63-64% on the same data. In order to improve on this, we have collected several hundred additional abstracts by querying the MedLine database. After pre-processing, this yields more than a hundred thousand (unlabelled) candidates that we may use with transductive inference and/or Fisher kernels.

8 Conclusion

In this paper, we presented a comparison between two state-of-the-art methods to combine labelled and unlabelled data: Fisher kernels and transductive inference. Our experimental results suggest that both method are able to yield a sizeable improvement in performance. For example transductive learning yields performance similar to inductive learning with only about a quarter of the data. These results are very encouraging for tasks where annotation is costly while unannotated data is easy to obtain, like our task of biological entity recognition. In addition, it provides a way to benefit from the availability of large electronic databases in order to automatically extract knowledge.

9 Acknowledgement

We thank Anne Schiller, Ágnes Sandor and Violaine Pillet for help with the data and related experimental results. This research was supported by the European Commission under the KerMIT project no. IST-2001-25431 and the French Ministry of Research under the BioMIRE project, grant 00S0356.

References

M. E. Califf, editor. 1999. *Proc. AAAI Workshop on Machine Learning for Information Extraction.* AAAI Press.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proc. ISMB'99.*

A. Gammerman, V. Vovk, and V. Vapnik. 1998. Learning by transduction. In Cooper and Morla, eds, *Proc. Uncertainty in Artificial Intelligence*, pages 145–155. Morgan Kaufmann.

Eric Gaussier, Cyril Goutte, Kris Popat, and Francine Chen. 2002. A hierarchical model for clustering and categorising documents. In Crestani, Girolami, and van Rijsbergen, eds, *Advances in Information Retrieval—Proc. ECIR'02*, pages 229–247. Springer.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proc. Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann.

Thomas Hofmann. 2000. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In *NIPS*12*, page 914. MIT Press.

Tommi S. Jaakkola and David Haussler. 1999. Exploiting generative models in discriminative classifiers. In *NIPS*11*, pages 487–493. MIT Press.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machine. In Bratko and Dzeroski, eds, *Proc. ICML'99*, pages 200–209. Morgan Kaufmann.

M. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman. 2000. Using blast for identifying gene and protein names in journal articles. *Gene.*

Y. Ohta, Y. Yamamoto, T. Okazaki, I. Uchiyama, and T. Takagi. 1997. Automatic constructing of knowledge base from biological papers. In *Proc. ISMB'97.*

D. Proux, F. Reichemann, and L. Julliard. 2000. A pragmatic information extraction strategy for gathering data on genetic interactions. In *Proc. ISMB'00.*

Bernhard Schölkopf and Alexander J. Smola. 2002. *Learning with Kernels.* MIT Press.

J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Caroll. 2000. Automatic extraction of protein interactions from scientific abstracts. In *Proc. PSB 2000.*

Kristina Toutanova, Francine Chen, Kris Popat, and Thomas Hofmann. 2001. Text classification in a hierarchical mixture model for small training sets. In *Proc. ACM Conf. Information and Knowledge Management.*

Vladimir N. Vapnik. 1998. *Statistical Learning Theory.* Wiley.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proc. 22nd ACM SIGIR*, pages 42–49.