Bootstrapping and Evaluating Named Entity Recognition in the Biomedical Domain

Andreas Vlachos
Computer Laboratory
University of Cambridge
Cambridge, CB3 0FD, UK
av308@cl.cam.ac.uk

Caroline Gasperin
Computer Laboratory
University of Cambridge
Cambridge, CB3 0FD, UK
cvg20@cl.cam.ac.uk

Abstract

We demonstrate that bootstrapping a gene name recognizer for FlyBase curation from automatically annotated noisy text is more effective than fully supervised training of the recognizer on more general manually annotated biomedical text. We present a new test set for this task based on an annotation scheme which distinguishes gene names from gene mentions, enabling a more consistent annotation. Evaluating our recognizer using this test set indicates that performance on unseen genes is its main weakness. We evaluate extensions to the technique used to generate training data designed to ameliorate this problem.

1 Introduction

The biomedical domain is of great interest to information extraction, due to the explosion in the amount of available information. In order to deal with this phenomenon, curated databases have been created in order to assist researchers to keep up with the knowledge published in their field (Hirschman et al., 2002; Liu and Friedman, 2003). The existence of such resources in combination with the need to perform information extraction efficiently in order to promote research in this domain, make it a very interesting field to develop and evaluate information extraction approaches.

Named entity recognition (NER) is one of the most important tasks in information extraction. It has been studied extensively in various domains, including the newswire (Tjong Kim Sang and De Meulder, 2003) domain and more recently the biomedical domain (Blaschke et al., 2004; Kim et al., 2004). These shared tasks aimed at evaluating fully supervised trainable systems. However, the limited availability of annotated material in most domains, including the biomedical, restricts the application of such methods. In order to circumvent this obstacle several approaches have been presented, among them active learning (Shen et al., 2004) and rule-based systems encoding domain specific knowledge (Gaizauskas et al., 2003).

In this work we build on the idea of bootstrapping, which has been applied by Collins & Singer (1999) in the newsire domain and by Morgan et al. (2004) in the biomedical domain. This approach is based on creating training material automatically using existing domain resources, which in turn is used to train a supervised named entity recognizer.

The structure of this paper is the following. Section 2 describes the construction of a new test set to evaluate named entity recognition for *Drosophila* fly genes. Section 3 compares bootstrapping to the use of manually annotated material for training a supervised method. An extension to the evaluation of NER appear in Section 4. Based on this evaluation, section 5 discusses ways of improving the performance of a gene name recognizer bootstrapped on FlyBase resources. Section 6 concludes the paper and suggests some future work.

2 Building a test set

In this section we present a new test set created to evaluate named entity recognition for *Drosophila* fly genes. To our knowledge, there is only one other test set built for this purpose, presented in Morgan et al. (2004), which was annotated by two annotators. The inter-annotator agreement achieved was 87% F-score between the two annotators, which according to the authors reflects the difficulty of the task.

Vlachos et al (2006) evaluated their system on both versions of this test set and obtained significantly different results. The disagreements between the two versions were attributed to difficulties in applying the guidelines used for the annotation. Therefore, they produced a version of this dataset resolving the differences between these two versions using revised guidelines, partially based on those developed for ACE (2004). In this work, we applied these guidelines to construct a new test set, which resulted in their refinement and clarification.

The basic idea is that *gene names* (<*gn*>) are annotated in any position they are encountered in the text, including cases where they are not referring to the actual gene but they are used to refer to a different entity. Names of gene families, reporter genes and genes not belonging to *Drosophila* are tagged as gene names too:

- the <*gn*>faf</*gn*> gene
- the <*gn*>Toll</*gn*> protein
- the <*gn*>string</*gn*>-<*gn*>LacZ</*gn*> reporter genes

In addition, following the ACE guidelines, for each *gene name* we annotate the shortest surrounding noun phrase. These noun phrases are classified further into *gene mentions* (<*gm*>) and *other mentions* (<*om*>), depending on whether the mentions refer to an actual gene or not respectively. Most of the times, this distinction can be performed by looking at the head noun of the noun phrase:

- <*gm*>the <*gn*>faf</*gn*> gene</*gm*>
- <*om*>the <*gn*>Reaper</*gn*> protein</*om*>

However, in many cases the noun phrase itself is not sufficient to classify the mention, especially when the mention consists of just the gene name, because it is quite common in the biomedical literature to use a gene name to refer to a protein or to other gene products. In order to classify such cases, the annotators need to take into account the context in which the mention appears. In the following examples, the word of the context that enables us to make the distinction between *gene mentions* (<*gm*>) and *other mentions* is underlined:

- ... ectopic expression of <*gm*><*gn*>hth</*gn*></*gm*> ...
- ... transcription of <*gm*><*gn*>string</*gn*></*gm*> ...
- ... <*om*><*gn*>Rols7</*gn*></*om*> localizes ...

It is worth noticing as well that sometimes more than one *gene name* may appear within the same noun phrase. As the examples that follow demonstrate, this enables us to annotate consistently cases of coordination, which is another source of disagreement (Dingare et al., 2004):

- <*gm*><*gn*>male-specific lethal-1</*gn*>, <*gn*>-2</*gn*> and <*gn*>-3</*gn*> genes</*gm*>

The test set produced consists of the abstracts from 82 articles curated by FlyBase[1]. We used the tokenizer of RASP[2] (Briscoe and Carroll, 2002) to process the text, resulting in 15703 tokens. The size and the characteristics of the dataset is comparable with that of Morgan et al (2004) as it can be observed from the statistics of Table 1, except for the number of non-unique gene-names. Apart from the different guidelines, another difference is that we used the original text of the abstracts, without any post-processing apart from the tokenization. The dataset from Morgan et al. (2004) had been stripped from all punctuation characters, e.g. periods and commas. Keeping the text intact renders this new dataset more realistic and most importantly it allows the use of tools that rely on this information, such as syntactic parsers.

The annotation of *gene names* was performed by a computational linguist and a FlyBase curator. We estimated the inter-annotator agreement in two

[1] www.flybase.net

[2] http://www.cogs.susx.ac.uk/lab/nlp/rasp/

	Morgan et al.	new dataset
abstracts	86	82
tokens	16779	15703
gene-names	1032	629
unique gene-names	347	326

Table 1: Statistics of the datasets

ways. First, we calculated the F-score achieved between them, which was 91%. Secondly, we used the Kappa coefficient (Carletta, 1996), which has become the standard evaluation metric and the score obtained was 0.905. This high agreement score can be attributed to the clarification of what *gene name* should capture through the introduction of *gene mention* and *other mention*. It must be mentioned that in the experiments that follow in the rest of the paper, only the *gene names* were used to evaluate the performance of bootstrapping. The identification and the classification of mentions is the subject of ongoing research.

The annotation of mentions presented greater difficulty, because computational linguists do not have sufficient knowledge of biology in order to use the context of the mentions whilst biologists are not trained to identify noun phrases in text. In this effort, the boundaries of the mentions where defined by the computational linguist and the classification was performed by the curator. A more detailed description of the guidelines, as well as the corpus itself in IOB format are available for download[3].

3 Bootstrapping NER

For the bootstrapping experiments presented in this paper we employed the system developed by Vlachos et al. (2006), which was an improvement of the system of Morgan et al. (2004). In brief, the abstracts of all the articles curated by FlyBase were retrieved and tokenized by RASP (Briscoe and Carroll, 2002). For each article, the gene names and their synonyms that were recorded by the curators were annotated automatically on its abstract using longest-extent pattern matching. The pattern matching is flexible in order to accommodate capitalization and punctuation variations. This process resulted in a large but noisy training set, consisting of 2,923,199 tokens and containing 117,279 gene names, 16,944 of which are unique. The abstracts used in the test set presented in the previous section were excluded. We used them though to evaluate the performance of the training data generation process and the results were 73.5% recall, 93% precision and 82.1% F-score.

Training	Recall	Precision	F-score
std	75%	88.2%	81.1%
std-enhanced	76.2%	87.7%	81.5%
BioCreative	35.9%	37.4%	36.7%

Table 2: Results using Vlachos et al. (2006) system

This material was used to train the HMM-based NER module of the open-source toolkit LingPipe[4]. The performance achieved on the corpus presented in the previous section appears in Table 2 in the row "std". Following the improvements suggested by Vlachos et al. (2006), we also re-annotated as gene-names the tokens that were annotated as such by the data generation process more than 80% of the time (row "std-enhanced"), which slightly increased the performance.

In order to assess the usefulness of this bootstrapping method, we evaluated the performance of the HMM-based tagger if we trained it on manually annotated data. For this purpose we used the annotated data from BioCreative-2004 (Blaschke et al., 2004) task 1A. In that task, the participants were requested to identify which terms in a biomedical research article are gene and/or protein names, which is roughly the same task as the one we are dealing with in this paper. Therefore we would expect that, even though the material used for the annotation is not drawn from the exact domain of our test data (FlyBase curated abstracts), it would still be useful to train a system to identify gene names. The results in Table 2 show that this is not the case. Apart from the domain shift, the deterioration of the performance could also be attributed to the different guidelines used. However, given that the tasks are roughly the same, it is a very important result that manually annotated training material leads to so poor performance, compared to the performance

[3] www.cl.cam.ac.uk/users/av308/Project_Index/node5.html

[4] http://www.alias-i.com/lingpipe/

achieved using automatically created training data. This evidence suggests that manually created resources, which are expensive, might not be useful even in slightly different tasks than those they were initially designed for. Moreover, it suggests that the use of semi-supervised or unsupervised methods for creating training material are alternatives worth-exploring.

4 Evaluating NER

The standard evaluation metric used for NER is the F-score (Van Rijsbergen, 1979), which is the harmonic average of Recall and Precision. It is very successful and popular, because it penalizes systems that underperform in any of these two aspects. Also, it takes into consideration the existence multi-token entities by rewarding systems able to identify the entity boundaries correctly and penalizing them for partial matches. In this section we suggest an extension to this evaluation, which we believe is meaningful and informative for trainable NER systems.

Two are the main expectations from trainable systems. The first one is that they will be able to identify entities that they have encountered during their training. This is not as easy as it might seem, because in many domains token(s) representing entity names of a certain type can appear as common words or representing an entity name of a different type. Using examples from the biomedical domain, "to" can be a gene name but it is also used as a preposition. Also gene names are commonly used as protein names, rendering the task of distinguishing between the two types non-trivial, even if examples of those names exist in the training data.

The second expectation is that trainable systems should be able to learn from the training data patterns that will allow it to generalize to unseen named entities. Important role in this aspect of the performance play the features that are dependent on the context and on observations on the tokens. The ability to generalize to unseen named entities is very significant because it is unlikely that training material can cover all possible names and moreover, in most domains, new names appear regularly.

A common way to assess these two aspects is to measure the performance on seen and unseen data separately. It is straightforward to apply this in tasks with token-based evaluation, such as part-of-speech tagging (Curran and Clark, 2003). However, in the case of NER, this is not entirely appropriate due to the existence of multi-token entities. For example, consider the case of the gene-name "head inhibition defective", which consists of three common words that are very likely to occur independently of each other in a training set. If this gene name appears in the test set but not in the training set, with a token-based evaluation its identification (or not) would count towards the performance on seen tokens if the tokens appeared independently. Moreover, a system would be rewarded or penalized for each of the tokens.

One approach to circumvent these problems and evaluate the performance of a system on unseen named entities, is to replace all the named entities of the test set with strings that do not appear in the training data, as in Morgan et al. (2004). There are two problems with this evaluation. Firstly, it alters the morphology of the unseen named entities, which is usually a source of good features to recognize them. Secondly, it affects the contexts in which the unseen named entities occur, which don't have to be the same as that of seen named entities.

In order to overcome these problems, we used the following method. We partitioned the correct answers and the recall errors according to whether the named entity at question have been encountered in the training data as a named entity at least once. The precision errors are partitioned in seen and unseen depending on whether the string that was incorrectly annotated as a named entity by the system has been encountered in the training data as a named entity at least once. Following the standard F-score definition, partially recognized named entities count as both precision and recall errors.

In examples from the biomedical domain, if "to" has been encountered at least once as a gene name in the data but an occurrence of in the test dataset is erroneously tagged as a gene name, this will count as a precision error on seen named entities. Similarly, if "to" has never been encountered in the training data as a gene name but an occurrence of it in the test dataset is erroneously tagged as a common word, this will count as a recall error on unseen named entities. In a multi-token example, if "head inhibition defective" is a gene name in the test dataset and it

	Recall	Precision	F-score	# entities
seen	95.9%	93.3%	94.5%	495
unseen	32.3%	63%	42.7%	134
overall	76.2%	87.7%	81.5%	629

Table 3: Extended evaluation

has been seen as such in the training data but the NER system tagged (erroneously) "head inhibition" as a gene name (which is not the training data), then this would result in a recall error on seen named entities and a precision error on unseen named entities.

5 Improving performance

Using this extended evaluation we re-evaluated the named entity recognition system of Vlachos et al. (2006) and Table 3 presents the results. The big gap in the performance on seen and unseen named entities can be attributed to the highly lexicalized nature of the algorithm used. Tokens that have not been seen in the training data are passed on to a module that classifies them according to their morphology, which given the variety of gene names and their overlap with common words is unlikely to be sufficient. Also, the limited window used by the tagger (previous label and two previous tokens) does not allow the capture of long-range contexts that could improve the recognition of unseen gene names.

We believe that this evaluation allows fair comparison between the data generation process that creating the training data and the HMM-based tagger. This comparison should take into account the performance of the latter only on seen named entities, since the former is applied only on those abstracts for which lists of the genes mentioned have been compiled manually by the curators. The result of this comparison is in favor of the HMM, which achieves 94.5% F-score compared to 82.1% of the data generation process, mainly due to the improved recall (95.9% versus 73.5%). This is a very encouraging result for bootstrapping techniques using noisy training material, because it demonstrates that the trained classifier can deal efficiently with the noise inserted.

From the analysis performed in this section, it becomes obvious that the system is rather weak in identifying unseen gene names. The latter contribute 31% of all the gene names in our test dataset, with respect to the training data produced automatically to train the HMM. Each of the following subsections describes different ideas employed to improve the performance of our system. As our baseline, we kept the version that uses the training data produced by re-annotating as gene names tokens that appear as part of gene names more than 80% of times. This version has resulted in the best performance obtained so far.

5.1 Substitution

A first approach to improve the overall performance is to increase the coverage of gene names in the training data. We noticed that the training set produced by the process described earlier contains 16944 unique gene names, while the dictionary of all gene names from FlyBase contains 97227 entries. This observation suggests that the dictionary is not fully exploited. This is expected, since the dictionary entries are obtained from the full papers while the training data generation process is applied only to their abstracts which are unlikely to contain all of them.

In order to include all the dictionary entries in the training material, we substituted in the training dataset produced earlier each of the existing gene names with entries from the dictionary. The process was repeated until each of the dictionary entries was included once in the training data. The assumption that we take advantage of is that gene names should appear in similar lexical contexts, even if the resulting text is nonsensical from a biomedical perspective. For example, in a sentence containing the phrase "the sws mutant", the immediate lexical context could justify the presence of any gene name in the place "sws", even though the whole sentence would become untruthful and even incomprehensible. Although through this process we are bound to repeat errors of the training data, we expect the gains from the increased coverage to alleviate their effect. The resulting corpus consisted of 4,062,439 tokens containing each of the 97227 gene names of the dictionary once. Training the HMM-based tagger with this data yielded 78.3% F-score (Table 4, row "sub"). 438 out of the 629 genes of the test set were seen in the training data.

The drop in precision exemplifies the importance

Training	Recall	Precision	F-score	cover
bsl	76.2%	87.7%	81.5%	69%
sub	73.6%	83.6%	78.3%	69.6%
bsl+sub	82.2%	83.4%	82.8%	79%

Table 4: Results using substitution

Training	Recall	Precision	F score	unseen F score
bsl	76.2%	87.7%	81.5%	42.7%
bsl-excl	80.8%	81.1%	81%	51.3%

Table 5: Results excluding sentences without entities

of using naturally occurring training material. Also, 59 gene names that were annotated in the training data due to the flexible pattern matching are not included anymore since they are not in the dictionary, which explains the drop in recall. Given these observations, we trained HMM-based tagger on both versions of the training data, which consisted of 5,527,024 tokens, 218,711 gene names, 106,235 of which are unique. The resulting classifier had seen in its training data 79% of the gene names in the test set (497 out of 629) and it achieved 82.8% F-score (row "bsl+sub" in Table 4). It is worth pointing out that this improvement is not due to ameliorating the performance on unseen named entities but due to including more of them in the training data, therefore taking advantage of the high performance on seen named entities (93.7%). Direct comparisons between these three versions of the system on seen and unseen gene names are not meaningful because the separation in seen and seen gene names changes with the the genes covered in the training set and therefore we would be evaluating on different data.

5.2 Excluding sentences not containing entities

From the evaluation of the dictionary based tagger in Section 3 we confirmed our initial expectation that it achieves high precision and relatively low recall. Therefore, we anticipate most mistakes in the training data to be unrecognized gene names (false negatives). In an attempt to reduce them, we removed from the training data sentences that did not contain any annotated gene names. This process resulted in keeping 63,872 from the original 111,810 sentences. Apparently, such processing would remove many correctly identified common words (true negatives), but given that the latter are more frequent in our data we expect it not to have significant impact. The results appear in Table 5.

In this experiment, we can compare the performances on unseen data because the gene names that were included in the training data did not change. As we expected, the F-score on unseen gene names rose substantially, mainly due to the improvement in recall (from 32.3% to 46.2%). The overall F-score deteriorated, which is due to the drop in precision. An error analysis showed that most of the precision errors introduced were on tokens that can be part of gene names as well as common words, which suggests that removing from the training data sentences without annotated entities, deprives the classifier from contexts that would help the resolution of such cases. Still though, such an approach could be of interest in cases where we expect a significant amount of novel gene names.

5.3 Filtering contexts

The results of the previous two subsections suggested that improvements can be achieved through substitution and exclusion of sentences without entities, attempting to include more gene names in the training data and exclude false negatives from them. However, the benefits from them were hampered because of the crude way these methods were applied, resulting in repetition of mistakes as well as exclusion of true negatives. Therefore, we tried to filter the contexts used for substitution and the sentences that were excluded using the confidence of the HMM based tagger.

In order to accomplish this, we used the "std-enhanced" version of the HMM based tagger to re-annotate the training data that had been generated automatically. From this process, we obtained a second version of the training data which we expected to be different from the original one by the data generation process, since the HMM based tagger should behave differently. Indeed, the agreement between the training data and its re-annotation by the HMM based tagger was 96% F-score. We estimated the entropy of the tagger for each token and for each sentence we calculated the average entropy over all

Training	Recall	Precision	F-score	cover
filter	75.6%	85.8%	80.4%	65.5%
filter-sub	80.1%	81%	80.6%	69.6%
filter-sub +bsl	83.3%	82.8%	83%	79%

Table 6: Results using filtering

its tokens. We expected that sentences less likely to contain errors would be sentences on which the two versions of the training data would agree and in addition the HMM based tagger would annotate with low entropy, an intuition similar to that of co-training (Blum and Mitchell, 1998). Following this, we removed from the dataset the sentences on which the HMM-based tagger disagree with the annotation of the data generation process, or it agreed with but the average entropy of their tokens was above a certain threshold. By setting this threshold at 0.01, we kept 72,534 from the original 111,810 sentences, which contained 61798 gene names, 11,574 of which are unique. Using this dataset as training data we achieved 80.4% F-score (row "filter" in Table 6). Even though this score is lower than our baseline (81.5% F-score), this filtered dataset should be more appropriate to apply substitution because it would contain fewer errors.

Indeed, applying substitution to this dataset resulted in better results, compared to applying it to the original data. The performance of the HMM-based tagger trained on it was 80.6% F-score (row "filter-sub" in Table 6) compared to 78.3% (row "sub" in Table 4). Since both training datasets contain the same gene names (the ones contained in the FlyBase dictionary), we can also compare the performance on unseen data, which improved from 46.7% to 48.6%. This improvement can be attributed to the exclusion of some false negatives from the training data, which improved the recall on unseen data from 42.9% to 47.1%. Finally, we combined the dataset produced with filtering and substitution with the original dataset. Training the HMM-based tagger on this dataset resulted in 83% F-score, which is the best performance we obtained.

6 Conclusions - Future work

In this paper we demonstrated empirically the efficiency of using automatically created training material for the task of *Drosophila* gene name recognition by comparing it with the use of manually annotated material from the broader biomedical domain. For this purpose, a test dataset was created using novel guidelines that allow more consistent manual annotation. We also presented an informative evaluation of the bootstrapped NER system that revealed that indicated its weakness in identifying unseen gene names. Based on this result we explored ways to improve its performance. These included taking fuller advantage of the dictionary of gene names from FlyBase, as well as filtering out likely mistakes from the training data using confidence estimations from the HMM-based tagger.

Our results point out some interesting directions for research. First of all, the efficiency of bootstrapping calls for its application in other tasks for which useful domain resources exist. As a complement task to NER, the identification and classification of the mentions surrounding the gene names should be tackled, because it is of interest to the users of biomedical IE systems to know not only the gene names but also whether the text refers to the actual gene or not. This could also be useful to anaphora resolution systems. Future work for bootstrapping NER in the biomedical domain should include efforts to incorporate more sophisticated features that would be able to capture more abstract contexts. In order to evaluate such approaches though, we believe it is important to test them on full papers which present greater variety of contexts in which gene names appear.

Acknowledgments

The authors would like to thank Nikiforos Karamanis and the FlyBase curators Ruth Seal and Chihiro Yamada for annotating the dataset and their advice in the guidelines. We would like also to thank MITRE organization for making their data available to us and in particular Alex Yeh for the BioCreative data and Alex Morgan for providing us with the dataset used in Morgan et al. (2004). The authors were funded by BBSRC grant 38688 and CAPES award from the Brazilian Government.

References

ACE. 2004. Annotation guidelines for entity detection and tracking (EDT).

Christian Blaschke, Lynette Hirschman, and Alexander Yeh, editors. 2004. *Proceedings of the BioCreative Workshop*, Granada, March.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT 1998*.

E. J. Briscoe and J. Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on EMNLP and VLC*.

J. Curran and S. Clark. 2003. Investigating gis and smoothing for maximum entropy taggers. In *Proceedings of the 11th Annual Meeting of the European Chapter of the Association for Computational Linguistics*.

S. Dingare, J. Finkel, M. Nissim, C. Manning, and C. Grover. 2004. A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. In *The 2004 BioLink meeting at ISMB*.

R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willet. 2003. Protein structures and information extraction from biological texts: The "PASTA" system. *BioInformatics*, 19(1):135–143.

L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu. 2002. Accomplishments and challenges in literature data mining for biology. *Bioinformatics*, 18(12):1553–1561.

J. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier, editors. 2004. *Proceedings of JNLPBA, Geneva*.

H. Liu and C. Friedman. 2003. Mining terminological knowledge in large biomedical corpora. In *Pacific Symposium on Biocomputing*, pages 415–426.

A. A. Morgan, L. Hirschman, M. Colosimo, A. S. Yeh, and J. B. Colombe. 2004. Gene name identification and normalization using a model organism database. *J. of Biomedical Informatics*, 37(6):396–410.

D. Shen, J. Zhang, J. Su, G. Zhou, and C. L. Tan. 2004. Multi-criteria-based active learning for named entity recongition. In *Proceedings of ACL 2004*, Barcelona.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

C. J. Van Rijsbergen. 1979. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow.

A. Vlachos, C. Gasperin, I. Lewin, and T. Briscoe. 2006. Bootstrapping the recognition and anaphoric linking of named entities in drosophila articles. In *Proceedings of PSB 2006*.