Adapting a Relation Extraction Pipeline for the BioCreAtIvE II Tasks

Claire Grover[1] grover@inf.ed.ac.uk
Barry Haddow[1] bhaddow@inf.ed.ac.uk
Ewan Klein[1] ewan@inf.ed.ac.uk
Michael Matthews[1] m.matthews@ed.ac.uk
Leif Arda Neilsen[1] lnielsen@inf.ed.ac.uk
Richard Tobin[1] richard@inf.ed.ac.uk
Xinglong Wang[1] xwang@inf.ed.ac.uk

[1] School of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh, EH8 9LW, Scotland

Abstract

The Second BioCreAtIvE Challenge provided an ideal opportunity to evaluate biomedical NLP techniques. Prior to the Challenge, an information extraction pipeline was developed to extract entities and relations relevant to the biomedical domain, and to normalise the entities to appropriate ontologies. With minimal effort, the pipeline was adapted to work with the BioCreAtIvE data and achieved results that appear competitive with existing state-of-the-art systems.

Keywords: biomedical NLP, relation extraction, named entity recognition, term identification

1 Introduction

The team 6 (T6) submissions for BioCreAtIvE II were based on research carried out as part of the TXM programme, a three year project aimed at producing NLP tools to assist in the curation of biomedical research papers. The principal product of this project is an information extraction pipeline, designed to extract entities and relations relevant to the biomedical domain, and to normalise the entities to appropriate ontologies. The submissions for the BioCreAtIvE II protein-protein interaction subtasks (IPS, ISS and IAS) used the output of the pipeline directly, whilst the submissions for the GM and GN tasks used techniques developed during the implementation of the pipeline. It was found that the TXM information extraction pipeline could be used without modification on the BioCreAtIvE II data, and appeared to maintain a similar level of performance as on the TXM test data.

For the phase 1 release of the TXM pipeline, the focus was on the recognition of protein mentions, protein-protein interactions (PPIs) and the normalisation of the proteins to a RefSeq-derived wordlist. In order to render the pipeline easily adaptable to other domains, machine learning approaches were favoured, and consequently a large quantity of annotated data was produced to train the system and to test its performance.

In Section 2, the information extraction pipeline is described, as well as the methods used in each of the five T6 submissions. Section 2 provides a brief analysis of the results for each submission, with an attempt to identify the major sources of error.

2 Methods

2.1 The TXM Information Extraction Pipeline

The TXM pipeline consists of a series of natural language processing tools, integrated within the LT-XML2 architecture.[1] In order to train and test the pipeline, we used a corpus of 151 full-texts and 749 abstracts which had been selected from PubMed and PubMedCentral as containing experimentally determined protein-protein interactions. The corpus was annotated by trained biologists for proteins and related entities, protein normalisations (to an in-house wordlist derived from RefSeq) and protein-protein interactions. Around 80% of the documents were used for training and optimising the pipeline, while the other 20% were held back for testing.

The major components of the pipeline are as follows:

Preprocessing The preprocessing component comprises tokenisation, sentence boundary detection, lemmatisation, part-of-speech tagging, species word identification, abbreviation detection and chunking. The part-of-speech tagging uses the Curran and Clark maximum entropy Markov model tagger [2] trained on MedPost data [16], whilst the other preprocessing stages are all rule-based. We implemented tokenisation, sentence boundary detection, species word identification and chunking with the LT-XML2 tools. For abbreviation extraction, we used the Schwartz and Hearst abbreviation extractor [14] and for lemmatisation we employed `morpha` [12].

Named Entity Recognition In the pipeline, named entity recognition (NER) of proteins is performed using the Curran and Clark classifier [2], augmented with extra features tailored to the biomedical domain.

Term Normalisation The term normalisation task in the pipeline involves choosing the correct identifier for each protein mention in the text, where the identifiers are drawn from a lexicon based on RefSeq. A set of candidate identifiers is generated using hand-written fuzzy matching rules, from which a single best identifier is chosen using a machine-learning based species tagger, and a set of heuristics to break ties. The term normalisation component of the pipeline was not used directly in the BioCreAtIvE II tasks since they employ different protein lexicons.

Relation Extraction To find the PPI mentions in the text, we built a maximum entropy relation extractor trained using shallow linguistic features [13]. The features include context words, parts-of-speech, chunk information, interaction words and interaction patterns culled from the literature. The relation extractor examines each pair of proteins mentioned in the text, and occurring less than a configurable number of sentences apart, and classifies them as being in an interaction or not. Whilst the relation extractor can theoretically recognise both inter-sentential and intra-sentential relations, since both types of candidate relations are considered, in practice very few inter-sentential relations are correctly recognised. Only around 5% of annotated relations are inter-sentential, and it is likely that using exactly the same techniques as on the intra-sentential relations is not optimal, especially since many of the inter-sententials use coreferences. The detection of inter-sentential relations is the subject of ongoing research.

In the remainder of this section, we will describe how this pipeline was deployed for carrying out the T6 submissions.

[1] `http://www.ltg.ed.ac.uk/software/xml/`

2.2 Gene Mention Task

To address the Gene Mention (GM) task, T6 employed two different machine learning methods using similar feature sets. Runs 1 and 3 used conditional random fields (CRF) [8], whilst run 2 used a bidirectional maximum entropy Markov model (BMEMM) [19].

Both CRF and BMEMM are methods for labelling sequences of words which model conditional probabilities so that a wide variety of possibly inter-dependent features can be used. The named entity recognition problem is represented as a sequential word tagging problem using the BIO encoding, as in CoNLL 2003 [18]. In BMEMM, a log-linear feature-based model represents the conditional probability of each tag, given the word and the preceding and succeeding tags. In CRF, by contrast, the conditional probability of the whole sequence of tags (in one sentence), given the words, is represented using a log-linear model. Both methods have been shown to give state-of-the-art performance in sequential labelling tasks such as chunking, part-of-speech-tagging and named entity recognition [10, 11, 15, 19]. The CRF tagger was implemented with CRF++[2] and the BMEMM tagger was based on Zhang Le's MaxEnt Toolkit.[3]

GM Preprocessing Before training or tagging the documents with the machine learner, we passed them through the preprocessing stages of the TXM pipeline (see Section 2.1).

GM Features For the machine learners, we extracted the following features for each word:

word The word itself is added as a feature, plus the four preceding words and four succeeding words, with their positions marked.

headword The headwords of noun and verb phrases are determined by the chunker, and, for all words contained in noun phrases, the head noun is added as a feature.

affix The affix feature includes all character n-grams with lengths between two and four (inclusive), and either starting at the first character, or ending at the last character of the word.

gazetteer The gazetteer features is calculated using an in-house list of protein synonyms derived from RefSeq. To add the gazetteer features to each word in a given sentence, the gazetteer is first used to generate a set of matched terms for the sentence, where each word is only allowed to be in one matched term and earlier starting, longer terms take precedence. The unigram gazetteer feature for each word has value either B, I or O, depending on whether the word is at the beginning, inside or outside of a gazetteer matched term. The bigram gazetteer feature is also added, and this is the concatenation of the previous and current word's gazetteer feature.

character For each of the regular expressions listed in Table 1, the character feature indicates whether or not the word matches the regular expression. These regular expressions were derived from lists published in previous work on biomedical and newswire NER [1, 2]. The length of the word is also included as a character feature.

postag This feature includes current word's part-of-speech tag and the POS tags for the two preceding and succeeding words. Also added are the bigram of the current and previous word's POS tag, and the trigram of the current and previous two words' POS tags.

wordshape The word shape feature consists of the word type feature of [2], a variant of this feature which only collapses runs of greater than two characters in a word, and bigrams of the word type feature.

abbreviation The abbreviation feature is applied to all abbreviations whose antecedent is found in the gazetteer.

[2] `http://chasen.org/~taku/software/CRF++/`

[3] `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`

Description	Regexp
Capitals, lower case, hyphen then digit	`[A-Z]+[a-z]*-[0-9]`
Capitals followed by digit	`[A-Z]{2,}[0-9]+`
Single capital	`[A-Z]`
Single Greek character	`\p{InGreek}`
Letters followed by digits	`[A-Za-z]+[0-9]+`
Lower case, hyphen then capitals	`[a-z]+-[A-Z]+`
Single digit	`[0-9]`
Two digits	`[0-9][0-9]`
Four digits	`[0-9][0-9][0-9][0-9]`
Two capitals	`[A-Z][A-Z]`
Three capitals	`[A-Z][A-Z][A-Z]`
Four capitals	`[A-Z]{4}`
Five or more capitals	`[A-Z]{5,}`
Digit then hyphen	`[0-9]+-`
All lower case	`[a-z]+`
All digits	`[0-9]+`
Nucleotide	`[AGCT]{3,}`
Capital, lower case then digit	`[A-Z][a-z]{2,}[0-9]`
Lower case, capitals then any	`[a-z][A-Z][A-Z].*`
Greek letter name	Match any Greek letter name
Roman digit	`[IVXLC]+`
Capital, lower, capital and any	`[A-Z][a-z][A-Z].*`
Contains digit	`.*[0-9].*`
Contains capital	`.*[A-Z].*`
Contains hyphen	`.*-.*`
Contains period	`.*\..*`
Contains punctuation	`.*\p{Punct}.*`
All digits	`[0-9]+`
All capitals	`[A-Z]+`
Is a personal title	`(Mr|Mrs|Miss|Dr|Ms)`
Looks like an acronym	`([A-Za-z]\.)+`

Table 1: The (Java) regular expressions used for the character feature in the GM task.

2.3 Gene Normalisation Task

ips The Gene Normalisation (GN) system was developed with genericity in mind. In other words, it can be ported to normalise other biological entities (e.g., disease types, experimental methods, etc) relatively easily, without requiring extensive knowledge of the new domain. The approach that was adopted combined a string similarity measure with machine learning techniques for disambiguation.

For GN, our system first preprocesses the documents (see Section 2.1) and then uses the gene mention NER component (see Section 2.2) to mark up gene and gene product entities in the documents. A fuzzy matcher then searches the gene lexicon provided and calculates scores of string similarity between the mentions and the entries in the lexicon using a measure similar to JaroWinkler [5, 6, 20].

The Jaro string similarity [5, 6] measure is based on the number and order of characters that are common to two strings. Given strings $s = a_1...a_k$ and $t = b_1...b_l$, define a character a_i in s to be *shared with* t if there is a b_j in t such that $b_j = a_i$ with $i - H \leq j \leq i + H$, where $H = \frac{min(|s|,|t|)}{2}$. Let $s' = a'_1...a'_{k'}$ be the characters in s which are shared with t (in the same order as they appear in s) and let $t' = b'_1...b'_{l'}$ be analogous. Now define a *transposition for* s', t' to be a position i such that

$a'_i \neq b'_j$. Let $T_{s',t'}$ be half the number of transpositions for s' and t'. The Jaro similarity metric for s and t is shown in Equation 1.

$$Jaro(s,t) = \frac{1}{3} \cdot \left(\frac{|s'|}{s} + \frac{|t'|}{t} + \frac{|s'| - T_{s',t'}}{|s'|} \right) \quad (1)$$

A variant of the Jaro measure due to Winkler [20] also uses the length P of the longest common prefix of s and t. It rewards strings which have a common prefix. Letting $P' = max(P, 4)$, it is defined as shown in Equation 2:

$$JaroWinkler(s,t) = Jaro(s,t) + \frac{P'}{10} \cdot (1 - Jaro(s,t)) \quad (2)$$

For the GN task, a variant of the JaroWinkler measure was employed, as shown in Equation 3, which uses different weighting parameters and takes into account the suffixes of the strings.

$$JaroWinkler'(s,t) = Jaro(s,t) + min(0.99, \frac{P'}{10} + \theta) \cdot (1 - Jaro(s,t)) \quad (3)$$

Here, $\theta = (\#\ CommonSuffix - \#\ DifferentSuffix)/lengthOfString$. The idea is not only to look at the common prefixes but also commonality and difference in string suffixes. A set of equivalent suffix pairs was defined, for example, the Arabic number 1 is defined as equivalent to the Roman number I. The number of common suffixes and the number of different suffixes (e.g., 1 and 2 or 1 and II would count as different suffixes) is counted, and strings with common suffixes are rewarded whilst those with different ones are penalised. The value is finally normalised by the length of the string.

At the end of the fuzzy-matching stage, each mention recognised by NER is associated with the single highest scoring match from the gene lexicon, where the score indicates the string similarity. Note that each match is associated with one or more identifiers (i.e., in cases where ambiguity occurs) from the gene lexicon.

The GN system collects all the gene identifiers, where every gene identifier is paired up with a set of features. These identifier-featureset pairs are used as training data to learn a model that predicts the most probable identifier out of a pool of candidates returned by the fuzzy matcher. Feature selection was manually carried out and simple features include the contextual text properties surrounding the mentions such as adjacent words, their part-of-speech tags, etc., and complex features such as the distance scores between the mentions in text and the matches returned by the fuzzy matcher. It turned out that the complex features are particularly helpful in terms of increasing the F_1 score.

In more detail, all the identifiers in a document found by the fuzzy matcher were collected, then the ones that are correct according to the answer file were used as positive examples and the others were used as negative ones. Each identifier was associated with a set of features as follows:

fuzzy-confidence Confidence scores[4] from the fuzzy matcher.

synonym-similarity The averaged confidence score of the similarity between all synonyms linked to the gene identifier and the match.

context-similarity The similarity between descriptions (ie., synonyms) associated with a gene identifier and all gene entities in the current document recognised by the NER. The similarity is calculated by two measures: Dice coefficient[5] and $tf * idf$.[6]

ner-confidence Confidence score generated by the NER tagger.

[4] Only those matches with confidence scores higher than 0.80 were considered.

[5] Dice coefficient is defined as twice the number of common terms in the two sets of tokens to compare, divided by the total number of tokens in both sets, ie., $Dice = \frac{2 * commonTerms}{\#\ of\ terms\ in\ set\ 1 + \#\ of\ terms\ in\ set\ 2}$.

[6] $tf * idf$ is defined as the product of *term frequency (tf)* and *inverse document frequency (idf)*. $tf_i = \frac{n_i}{\sum_k n_k}$, where n_i is the number of occurrences of the considered term and the denominator is the number of occurrences of all terms. $idf_i = log \frac{|D|}{|\{d : d \ni t_i|\}}$, where $|D|$ is the total number of documents and the denominator is the number of documents where the term t_i appears.

context Local features, including contextual words (±10),[7] lemmas (±4), POS tags (±2), species words (±10) and bigrams (±5).

length Length of the gene mention and length of the match.

With the positive and negative examples extracted, determining the correct normalisations becomes a standard machine learning task. A classifier using SVM^{light} [7] was trained on the examples extracted from the BioCreAtIvE II GN training data.

The documents were processed similarly for the testing. In detail, a document was first run through the NER tagger where all the potential entities were marked up. The fuzzy matcher then searched the gene lexicon and produced a list of candidate gene identifiers, which were associated with the features extracted from the context of the document and classified using the SVM model trained in the training stage. Finally, the positive identifiers predicted by the model were output as the correct normalisations of the document.

2.4 Interaction Article Subtask

The Interaction Article Subtask (IA) was treated as a standard document classification problem where abstracts were classified as CURATABLE if they contained curatable protein interaction information and NOT-CURATABLE otherwise. Document classification techniques typically use a bag-of-words approach which ignores the word order in the document. This approach was extended by using a 'bag-of-nlp' approach where in addition to words, a variety of features derived from the output of a natural language processing (NLP) pipeline were added to the bag. The classification was performed with SVM^{light} [7] using the linear kernel with the default parameters. The documents were ordered based on the output from the SVM classifier.

IA Preprocessing Before the documents were passed to the machine learner for training or classification, they were first passed through the the TXM pipeline (see Section 2.1). In addition, each of the named-entities and compound nouns in the document were marked as phrases.

IA Features The features extracted for each document are described below. Only features that occurred at least twice in the training data were used and each feature was given a binary weight. Each feature was converted to lowercase and words found in a custom stopword list were ignored. For each word and word stem, BACKOFF and BACKOFF-STEMMED versions were also calculated by converting all numbers to a single '#' symbol and removing all punctuation.

word The word itself.

word-backoff The BACKOFF version of the word.

bigram The bigrams of the BACKOFF feature. The bigrams were not allowed to cross sentence boundaries.

chunk The concatenation of the BACKOFF-STEMMED version of each word in a chunk up to a maximum of seven words.

phrase The concatenation of the BACKOFF-STEMMED version of each word in a phrase (one-word phrases were included).

phrase-bigram The bigrams of the PHRASE feature. All proteins were converted to the token NER-PROTEIN. The bigrams were not allowed to cross sentence boundaries.

chunk-headword-bigram The bigrams of the BACKOFF-STEMMED version of each headword of successive chunks. Chunks containing negative phrases (e.g., does not interact) were indicated by preceding the bigram with NEG.

[7] The numbers in parentheses denote the size of the context window.

chunk-headword-trigram The trigrams of the BACKOFF-STEMMED version of each headword of successive chunks. All proteins were converted to the token NERPROTEIN. Chunks containing negative phrases were indicated by preceding the trigram with NEG.
protein Added if the document contained at least one protein.
two-proteins Added if the document contained at least two unique proteins.
no-proteins Added if the document did not contain any proteins.
title-proteins Added if the document contained two unique proteins in the title.

2.5 Interaction Pair Subtask

The T6 Interaction Pair (IP) Subtask system made use of the TXM information extraction pipeline to identify mentions of protein-protein interactions (PPIs), together with additional components to normalise proteins to UNIPROT and to identify the curatable interactions from amongst the interaction mentions.

Data Preparation Two methods of data preparation were used. In runs 1 and 3, the supplied `pdftotext` converted files were converted to the XML input format required by the pipeline, essentially by just wrapping the text in <`text`> and <`document`> elements and removing illegal characters.[8] In run 2, however, the supplied HTML files were used, having been first run through an in-house HTML to XML converter.

PPI Extraction The named entity recognition and relation extraction stages of the pipeline (Section 2.1) were used to identify mentions of protein-protein interactions.

UniProt Normalisation Two approaches were used to assign UNIPROT identifiers to protein mentions, exact matching (in runs 1 and 2) and fuzzy matching (in run 3). In exact matching, the protein name in the text is compared against each protein synonym in the UNIPROT lexicon using a case-insensitive match, to obtain a list of possible identifiers. If no possible identifiers are found, and the protein name is the long or short form of an abbreviation identified by the abbreviation extractor, then the corresponding (short or long) form is also looked up in the lexicon. In order to filter the list of identifiers, each identifier is weighted according to how often its corresponding species name is mentioned in the text, with species name mentions closer to the protein mention receiving higher weights. The identifier with the highest weight is then chosen.

The fuzzy match protein normaliser uses a string distance measure (see Section 2.3) to find the set of protein names in the lexicon which are closest to the protein mention in the text. These distances are then weighted according to the species word mentions, as for exact matching, and the highest weighted identifier chosen.

Curation Filter The curation filter takes as its input the set of UNIPROT identifier pairs representing the interactions found in the text by the pipeline, with their UNIPROT normalisations, and outputs the set of normalised, curatable interactions. The filter was implemented with an SVM classifier (using SVM^{light} [7] with an RBF kernel), trained on the supplied training data, using the following set of features:

relation-count This feature counts the number of times that the interaction is mentioned in the document.

[8] These were ascii control characters inserted by `pdftotext`, which are not legal in XML. They were all removed except for ascii `0x0C`, which was converted to a double newline.

inter-sentential This indicates whether the majority of the mentions of the interaction are inter-sentential relations between proteins, or intra-sentential. As noted in Section 2.1, the relation extractor does not perform well on inter-sentential relations, so very few of these are predicted (only 15 in the training corpus).

relation-confidence Each interaction mention found by the pipeline has an associated confidence. The value of this feature is the maximum confidence assigned to an interaction's mentions.

position This feature specifies the relative position within the document of the first and last mentions of the interaction. In addition, the mean relative position of the interaction mentions is included, for each interaction.

species The species feature indicates whether the proteins in the proposed interaction have different species.

title This feature indicates whether the interaction is mentioned in the title.

normalisation-confidence When using the fuzzy-matched normalisations, this feature indicates how close a match has been found during normalisation of the protein mention.

As recommended in the IP Subtask instructions, any documents containing more than 30 interactions were excluded from the training set.

2.6 Interaction Sentence Subtask

The T6 Interaction Sentence Subtask system was identical to the system used for run 2 of the IP Subtask (see Section 2.5) with the addition of the following two steps:

Data Preparation The HTML to XML converter preserved a mapping between the HTML text and the sentences in the converted XML file.

Passage Selection The interaction mentions for each curatable interaction were sorted according to the confidence values associated with each mention and the sentences associated with the top five mentions were returned as the relevent passages.

3 Analysis

3.1 Gene Mention

As reported in Section 2.2, two different techniques were used for the gene mention task, conditional random fields (CRF) and bidirectional maximum entropy markov models (BMEMM). Runs 1 and 3 both used CRF (with different settings of the Gaussian prior) whilst run 2 used BMEMM, with all three runs using the same feature set. The results are shown in Table 2. The distribution of scores on the test

Run	Method	Heldout			Test		
		Precision	Recall	F_1	Precision	Recall	F_1
1	CRF	0.8594	0.8211	0.8398	0.8697	0.8255	0.8470
2	BMEMM	0.8597	0.7982	0.8278	0.8638	0.8041	0.8329
3	CRF	0.8463	0.8297	0.8379	0.8649	0.8248	0.8444

Table 2: Performance comparison for the Gene Mention task. In the heldout configuration, the system was trained on 80% of the data and tested on 20%, whilst in the test configuration the system was trained on all the training data and tested on the test data.

data matches that obtained during heldout testing on the training data, in that CRF outperformed BMEMM on F_1 (mainly due to higher recall) and the run 1 configuration was the best overall.

3.2 Gene Normalisation

As described in Section 2.3, we produced 3 runs for the Gene Normalisation task. The results are shown in Table 3.

Run	Method	Precision	Recall	F_1
1	ML Filter 1	0.767	0.601	0.674
2	ML Filter 2	0.767	0.606	0.677
3	Heuristics Filter	0.597	0.782	0.677

Table 3: Performance comparison for the Gene Normalisation task on the test data. The machine learning (ML) Filter 1 uses Dice measure to calculate the similarity between synonyms associated with the identifier and all entities detected by NER in the current document; while ML Filter 2 uses *tf*idf* for the same task. The Heuristics Filter simply chooses the identifier that has the lowest number.

The approach is not completely supervised because the training data constructed for a document does not necessarily contain all the correct identifiers as given in the answer file. The coverage of our fuzzy matcher is up to 88%, which is an upperbound for the recall of the GN system. The approach takes advantage of string similarity measures that are more generic than hand-coded knowledge when carrying out the fuzzy matching. Combined with machine learning techniques, the T6 system is more portable than some GN systems reported in previous work [4, 3].

3.3 Interaction Article Subtask

Table 4 compares results of a bag-of-words baseline system to the bag-of-nlp system. The baseline system uses only the *word* and *bigram* features but is otherwise the same as the bag-of-nlp system. The results are presented both for 5-fold cross-validation on the training set and for the test set.

System	5-Fold cross-validation					Test				
	AUC	Prec	Rec	F_1	Acc	AUC	Prec	Rec	F_1	Acc
baseline	0.9757	0.9452	0.9420	0.9436	0.9276	0.8188	0.6898	0.8480	0.7608	0.7333
bag-of-nlp	0.9777	0.9550	0.9474	0.9512	0.9374	0.8483	0.6994	0.8747	0.7773	0.7493

Table 4: Overall Results

Data Inconsistency The most obvious observation is the drop in performance from cross-validation to test. This can be partially explained by some inconsistencies between the training and test sets. When analysing the test set, it was noticed that 37 of the files were actually also present in the training set. Furthermore, 13 of these files had a different label in the test set than in the training set: in each of the differences a document that was labelled as a positive example in the training set was labelled as a negative example in the test set. This would explain why the precision has gone down more than the recall. In order to estimate the effect of these differences, the bag-of-nlp system was trained on all of the training documents with the exception of these 13 documents and then used to predict the class of the 13 files. In 12 of the 13 cases, the system predicted that the articles were positive examples and thus found to be incorrect in the final evaluation. If these 13 files had been labelled as positive in the test set, the precision would have risen from the reported 0.699 to 0.725. A manual examination of some of the files in question suggests that the abstracts do contain interactions, but it is difficult to

determine if the full text versions meet the standards for curation. Regardless, the differences between the labels in test and training raise concerns about how representative the test data is of the training data.

NLP Benefits The next observation is that the bag-of-nlp system does provide a small improvement over the baseline system. The NLP features are based largely on either the NER module or the chunker. In order to assess the relative contribution of each component, a lesion test was performed where the system was run without NER and then without the chunker. The results are presented in Table 5.

System	5-Fold cross-validation					Test				
	AUC	Prec	Rec	F_1	Acc	AUC	Prec	Rec	F_1	Acc
bag-of-nlp	0.9777	0.9550	0.9474	0.9512	0.9374	0.8483	0.6994	0.8747	0.7773	0.7493
no NER	0.9771	0.9498	0.9465	0.9482	0.9334	0.8277	0.6908	0.8640	0.7678	0.7387
no chunker	0.9779	0.9530	0.9471	0.9501	0.9359	0.8412	0.6956	0.8773	0.7759	0.7467

Table 5: Benefits from NLP

The results indicate that the NER module is more useful than the chunker. Overall, however, the contribution from NLP is less significant than one would hope and less than reported in previous work [9]. One possibility is that since the baseline system already performs at a very high level, the contributions of imperfect NLP are not as effective. This is supported by the fact that the relation extraction component, which has an F_1 score of less than 0.50, actually hurt system performance and was therefore not included in the final bag-of-nlp system. In the future, it would be useful to perform experiments on a dataset that has been annotated with both document classes and linguistic information to determine the benefits of human-level NLP on document classification. This would at least provide an upper bound for how much improvement could be provided by NLP.

3.4 Interaction Pair Subtask

For the submissions to the Interaction Pair (IP) Subtask, exact matching normalisation was used in runs 1 and 2, and fuzzy matching in run 3, whilst the PDFconverted files were used in runs 1 and 3, and the HTML converted files in run 2. During cross-validation testing on the training set, the configuration in run 1 achieved the highest score, followed by the run 2 configuration, and then the run 3 configuration. However, as can be see in Table 6, the run 3 configuration achieved the highest score on the test data.

Run	Filetype	Normaliser	10-fold cross validation			Test		
			Precision	Recall	F_1	Precision	Recall	F_1
1	PDF	exact	0.2687	0.1712	0.2091	0.2302	0.1283	0.1648
2	HTML	exact	0.2574	0.1702	0.2049	0.2003	0.1204	0.1504
3	PDF	fuzzy	0.2354	0.1756	0.2011	0.2131	0.1496	0.1758

Table 6: Comparison of performance for different data file types and normalisers. The system was tested using 10-fold cross-validation on the training data, and on the test data.

Since the overall system for the IP Subtask comprised several different stages, it would be useful to gain some idea of the performance of each stage to see where improvements could be made. In the rest of this section, each component will be considered in turn to discuss how it contributes to the overall IP Subtask errors.

Named Entity Recognition (NER) When tested on the TXM blind test corpus, the NER component achieves an F_1 score of 78% on protein mentions. Within the IP Subtask, NER can cause both false negatives, if the NER component does not correctly recognise a protein that is involved in a curatable interaction, and false positives, if the NER component incorrectly marks a non-protein as a protein, and that protein is then placed in an interaction and normalised by subsequent processing stages. The NER component can also make boundary errors, where it identifies a protein at the correct location but gets its boundaries wrong, making the task more difficult for the normaliser. There is no gold NER data available for the IP Subtask test documents, but an estimate of the recall of NER and normalisation combined can be obtained by counting the number of gold interactions in the IP Subtask test data where the system correctly identified and normalised both proteins in the interaction. Using the configuration in run 1 (exact match normaliser and pdf converted documents), both proteins were correctly identified in 43.86% of the gold interactions.

Relation Extraction (RE) The RE component, when tested on the TXM blind test corpus, using gold NE data, achieves an F_1 of about 45% on the identification of protein-protein interaction (PPI) mentions. Table 7 gives an upper bound on the recall of the RE component, in the context of the IP Subtask, by showing the counts of true positives obtained by considering all generated matches for all the protein pairs output by RE (note that the recall figure here is lower than the 43.86 mentioned in the previous paragraph, since the figures in the table only include those proteins which the RE component has predicted to be in interactions, whilst the 43.86 includes all proteins predicted by NER). The RE component can introduce false positives into the IP Subtask by identifying incorrect PPIs, which are then classed as curatable by the curation filter, and can introduce false negatives by missing mentions of curatable interactions. It is also possible that curatable interactions are not mentioned directly in the document, but are inferred from experimental descriptions, and so would never be detected by the RE component.

Normalisation In the normalisation component, a list of possible matches is generated for each protein mention using a string matching algorithm, and then this list of matches is reordered using the species information found in the text. The normalisation requirement in the IP Subtask complicated any error analysis, since the gold data (in the form of pairs of Uniprot identifiers) could not be matched directly with the text. Nevertheless, a measure of the recall of NER and normalisation combined was given above, and the effectiveness of the species-based disambiguation can be gauged from the results shown in Table 7. This table shows how the disambiguator reduces the number of false positives (obtained by pairing all matched normalisations for each predicted pair of interacting proteins) by about 3 orders of magnitude.

Curation Filtering Table 7 also illustrates the effectiveness of the machine learning based curation filter in removing false positives. In general it achieves around a 10-fold reduction in false positives, whilst removing around a third of true positives. The threshold of the filter could be adjusted to favour precision or recall, but for the IP Subtask submission it was optimised to give the highest possible F_1 when cross-validating on the training set.

In summary, the relation extractor and the normaliser seem to be the main areas where improvements could be made. The relation extractor achieves an F_1 of 45% on PPI mentions in the TXM data, which compares well to the inter-annotator agreement (IAA) of 52% on this data, but is low in absolute terms. It should be emphasised that this score is on PPI mentions, and since a curatable interaction may be mentioned several times, or perhaps not explicitly mentioned at all, it is not clear exactly what effect the score on PPI mentions has on the IP Subtask.

The low IAA was a cause for concern within the TXM project and thus efforts were made improve it in the second round of annotation. This round was completed after the BioCreAtIvE II challenge

Filetype	Normaliser	Stage	TP	FP
PDF	exact	Generate matches	333 (29.46%)	1,121,979
		Disambiguate	223 (19.73%)	4,351
		Curation filter	145 (12.83%)	485
HTML	exact	Generate matches	314 (27.79%)	1,077,231
		Disambiguate	207 (18.32%)	3,939
		Curation filter	136 (12.04%)	543
PDF	fuzzy	Generate matches	449 (39.73%)	9,016,377
		Disambiguate	271 (23.98%)	8,069
		Curation filter	169 (14.96%)	624

Table 7: Comparison of performance on the IP Subtask test data before and after species-based disambiguation, and after curation filtering. The percentage of true positives (TP) is measured against the total number of gold interactions.

ended and employed several iterations of piloting the annotation and revising the guidelines before starting the annotation for real. The IAA on PPIs increased to an F_1 of 64.77%, a score which is still lower than might be hoped, but which is believed to accurately reflect the inherent difficulty of the task. Unfortunately, to the best of our knowledge, there are few published IAA figures from similar annotation tasks from other groups, making comparison with other work difficult.

As noted in Section 2.3, normalisation of proteins in biomedical text is a hard task, and the normalisation within the IP Subtask is especially hard as the species is not given in advance. From Table 7 it can be seen that disambiguation is a significant problem in normalisation, with up to 40% of correctly normalised pairs erroneously removed by the disambiguator.

3.5 Interaction Sentence Subtask

The preliminary results for the Interaction Sentence (IS) Subtask are shown in Table 8. As mentioned

Description	Value
No. eval. predicted passages	2,497
No. eval. unique passages	2,072
No. eval. matches to previously selected	147
No. eval. unique matches to previously selected	117
Fraction correct (best) from predicted passages	0.0589
Fraction correct (best) from unique passages	0.0565
Mean reciprocal rank of correct passages	0.5525

Table 8: IS Subtask Evaluation Summary

in the methods section, the passages selected for this system were derived from the output of run 2 of the IP Subtask system. The biggest drawback of this system is that the relation extraction module is trained to identify all protein-protein interactions and not just curatable interactions. Therefore, the confidences that are used to rank the passages do not take into account the curatability of the sentence, only the degree of certainty as to whether they represent a protein-protein interaction. It would be possible in the future to rerank these passages based on the training data provided as part of the ISS task.

A further drawback is that the IP Subtask system was optimised to correctly normalise the protein mentions. However, for the IS Subtask, it was not critical to identify the correct normalisations, but

rather just the correct passages. Thus, the system could potentially be improved by skipping the disambiguation step. Table 7 indicates that more than 100 correct interactions, over 30% of the total, were incorrectly filtered out during the disambiguation stage. Though it is difficult to determine how removing this stage would effect the IS Subtask scores, it does suggest that some improvement could be made.

4 Conclusions

For the PPI subtasks (IP, IS and IA), the information extraction pipeline developed for the TXM programme proved effective since it addressed related problems (identification of proteins and their interactions) and was trained on similar data to that used in BioCreAtIvE II. For the IP Subtask, the pipeline architecture was easily extended with two extra components (normalisation and curation filtering) specific to the requirements of the subtask, showing the flexibility of this architecture.

The approach to normalisation that we have adopted, based on a string distance measure and machine learning disambiguation, has the advantage that it should be more easily adaptable to other normalisation problems (e.g., tissues, cell-lines) than an approach based on manually created matching rules. Although better results may currently be obtained with rule-based methods, we believe that our proposed approach offers more promise for the future. Given that it is very hard to automatically predict the single correct identifier for a biomedical entity (such as a protein), it would be interesting to explore the relative merits of an approach which focuses on mimimizing the number of candidate identifiers, as compared to supplying the user with fuzzy matching tools to help search ontologies interactively.

The T6 approach to the IP Subtask involved trying to reconstruct curated information from interactions mentioned explicitly in the text; however, it is not known what proportion of curated data can be obtained this way. In other words, are all or most curatable interactions mentioned explicitly in the text as an interaction between two mentioned proteins? A recent paper [17] showed that a significant proportion of facts in the MUC evaluations are distributed across several sentences, and similar results seem likely to apply in the biomedical domain. While the low overall scores in the IP Subtask show that NLP techniques are not yet ready to replace manual curation, they may be nevertheless able to aid curators in their work, or be used to produce large volume, noisy data which is of benefit to biologists.

5 Acknowledgements

The TXM pipeline on which this system was based was carried out as part of a joint project with Cognia (`http://www.cognia.com`), supported by the Text Mining Programme of ITI Life Sciences Scotland (`http://www.itilifesciences.com`). The authors would also like to thank Beatrice Alex, Mijail Kabadjov and Stuart Roebuck for all their assistance during the development of the system and the preparation of this manuscript.

References

[1] N. Collier and K. Takeuchi. Comparison of character-level and part of speech features for name recognition in biomedical texts. *Journal of Biomedical Informatics*, 37(6):423–435, 2004.

[2] J. R. Curran and S. Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-2003*, pages 164–167, 2003.

[3] H. Fang, K. Murphy, Y. Jin, J. S. Kim, and P. S. White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of BioNLP'06*, New York, USA, 2006.

[4] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck. ProMiner: Organism-specific protein name detection using approximate string matching. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.

[5] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84:414–420, 1989.

[6] M. A. Jaro. Probabilistic linkage of large public health data files. *Statistical in Medicine*, 14:491–498, 1995.

[7] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1999.

[8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[9] M. Matthews. Improving biomedical text categorization with NLP. In *Proceedings of the SIGs, The Joint BioLINK-Bio-Ontologies Meeting, ISMB 2006*, pages 93–96, 2006.

[10] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 188–191. Edmonton, Canada, 2003.

[11] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6(Suppl1):S6, 2005.

[12] G. Minnen, J. Carroll, and D. Pearce. Robust, applied morphological generation. In *Proceedings of 1st International Natural Language Generation Conference (INLG'2000)*, 2000.

[13] L. A. Nielsen. Extracting protein-protein interactions using simple contextual features. In *Proceedings of the BioNLP workshop, HLT/NAACL 2006 - poster session*, pages 120–121, 2006.

[14] A. Schwartz and M. Hearst. Identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing*, pages 451–462, 2003.

[15] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In M. Hearst and M. Ostendorf, editors, *Proceedings of HTL-NAACL-2003*, pages 213–220, Edmonton, Canada, 2003.

[16] L. Smith, T. Rindflesch, and W. J. Wilbur. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321, 2004.

[17] M. Stevenson. Fact distribution in information extraction. *Language Resources and Evaluation*, 40(2):183–201, 2006.

[18] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147, 2003.

[19] Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 467–474, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[20] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistics of Income Division, Internal Revenue Service Publication R99/04, 1999.