Use of unsupervised word classes for entity recognition. Application to the detection of disorders in clinical reports

Maria Evangelia Chatzimina

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, Heraklion, Crete, GR-70013, Greece

Septembre 2013

University of Crete
Computer Science Department

Use of unsupervised word classes for entity recognition. Application to the detection of disorders in clinical reports

Thesis submitted by
Maria Evangelia Chatzimina
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Maria Evangelia Chatzimina

Committee approvals: _____
Pierre Zweigenbaum
Professor, Thesis Supervisor, Committee member

_____
Alexandre Allauzen
Assistant Professor, Committee member

_____
Thomas Lavergne
Assistant Professor, Committee member

Paris, September 2013

Περίληψη

Η επεξεργασία υπολογιστικής γλώσσας είναι ο κλάδος της επιστήμης των υπολογιστών που επικεντρώνεται στην ανάπτυξη συστημάτων που επιτρέπουν στους υπολογιστές να επικοινωνούν με τους χρήστες που χρησιμοποιούν φυσική γλώσσα. Πολλές τεχνικές της υπολογιστικής γλώσσας όπως αναγνώριση οντότητας, αποκοπή καταλήξεων, αναγνώριση του μέρους του λόγου και άλλες έχουν χρησιμοποιηθεί για την εξαγώγη πληροφοριών. Σε πολλες περιπτώσεις σημασιολογικές πληροφοριές χρησιμοποιούνται για να επεκτείνουν τις γνώσεις σχετικά με τα έγγραφα και για την βελτίωση της επίδοσης.

Υπάρχει ένα αυξανόμενο ενδιαφέρον για στρατηγικές υπολογιστικής γλώσσας που εφαρμόζονται στα κλινικά έγγραφα και οφείλεται στην αύξηση του αριθμού των ηλεκτρονικών εγγραφων στα πληροφοριακά συστήματα των νοσοκομείων. Η εξόρυξη βιοιατρικού κειμένου είναι ένα πεδίο έρευνας που αναφέρεται στην εξόρυξη κειμένου που έχει εφαρμοστεί σε ιατρικά κείμενα ή σε κείμενα του τομέα της βιοιατρικής.

Στην παρούσα εργασία παρουσιάζουμε μια μεθοδολογία που συνδυάζει αλγορίθμους με τεχνικές επιτηρούμενης και μη επιτηρούμενης μηχανικής μάθησης προκειμένου να συμβάλει στην αναγνώριση οντοτήτων σχετικών με κλινικά κείμενα. Η αναγνώριση οντοτήτων γίνεται βασή συστημάτων γνώσεων σημασιολογικών πόρων. Παρουσίαζουμε μια προσέγγιση όπου οι κατηγορίες λέξεων που λειτουργούν βάση των δεδομένων είσοδου αξιολογούνται και συγκρίνονται με συστημάτα γνώσεων σημασιολογικών πόρων όταν εισάγωνται ως χαρακτηριστικά σε ενα κατηγοριοποιητή Conditional Random Field (CRF). Εξετάζουμε διαφορετικές μεθόδους οι οποίες συνδυάζουν συστημάτα γνώσεων σημασιολογικών πόρων με κατηγορίες λέξεων που λειτουργούν βάση των δεδομένων είσοδου με σκοπό να βελτιώσουν την αναγνώριση οντοτήτων. Τα δεδομένα με γνώμονα σημασιολογικές κατηγορίες πέτυξαν αποτελέσματα με μικρές διαφορές σε σύγκριση με τα συστήματα γνώσεων σημασιολογικών πόρων. Η μελέτη μας κατέληξε στο συμπέρασμα οτι οι κατηγορίες λέξεων που λειτουργούν βάση των δεδομένων είσοδου προσθέτουν σημαντικές πληροφορίες και ειναι συμπληρωματικές με τα συστήματα γνώσεων σημασιολογικών πόρων.

Abstract

Natural language processing (NLP) is the branch of computer science focused on developing systems that allow computers to communicate with people using everyday language. Many NLP techniques, including stemming, part of speech tagging, named entity recognition, compound recognition, de-compounding, chunking, word sense disambiguation and others, have been used for information extraction. In many cases, semantic information is used to expand knowledge about documents and to improve performance.

There is an increasing interest in NLP strategies applied to clinical texts due to the increasing number of electronic documents in hospital information systems. Biomedical text mining is a research field on the edge of natural language processing and refers to text mining applied to clinical text or to the literature of the biomedical domain.

In this work, we present a methodology which combines unsupervised word classes with supervised machine learning methods in order to contribute to named entity recognition on clinical reports. Named entity recognition is performed generally by knowledge-based semantic resources. We present an approach where data-driven word classes are evaluated and compared with knowledge-based semantic classes when inserted as features in a Conditional Random Field (CRF) classifier. We examine different methods to combine data-driven word classes with knowledge-based semantic classes to improve named entity recognition. Data-driven semantic classes achieve results with small differences compared to knowledge-based semantic classes. Our case study concluded that data-driven word classes can add important information and are complementary with knowledge-based semantic classes.

Résumé

Le traitement automatique des langues (TAL) est la branche de l´informatique qui vise à développer des systèmes qui permettent aux ordinateurs de communiquer avec les humains en utilisant la langue naturelle. Le TAL est un élément clé dans la conception des programmes informatiques dont les instructions doivent être prêtes pour le traitement par les machines et compréhensibles pour les humains. Beaucoup de techniques du TAL, parmi lesquelles la racinisation, l´étiquetage en parties du discours, la reconnaissance d´entité nommées, la reconnaissance de composés, la décomposition, la segmentation en syntagmes, la désambiguïsation sémantique, ont été utilisées pour la classification de textes. Dans de nombreux cas, l´information sémantique est utilisées pour accroître la connaissance des documents et pour augmenter les performances. Dans un tel système, le TAL est utilisé pour apparier le contenu sémantique avec le document à classer.

On constate un intérêt croissant pour l´application des techniques du TAL à la littérature biomédicale en raison de l´augmentation du nombre de publications disponibles sous forme électronique. La fouille de texte en domaine biomédical est un champ de recherche lié au TAL qui renvoie à la fouille de texte appliquée aux textes et documents des domaines biomédicaux et de la biologie moléculaire.

Dans ce travail, nous présentons une tentative d´utilisation de clusters de mots non supervisés au moyen de méthodes par apprentissage supervisées afin d´effectuer une reconnaissance d´entités nommées dans des comptes rendus cliniques. La reconnaissance d´entités nommées est généralement réalisée sur la base de resources sémantiques et de connaissances d´expert. Nous présentons une approche où les clusters de mots sont utilisés par un classifieur à base de CRF de chaîne linéaire, puis évalués et comparés avec des classes sémantiques reposant sur des connaissances d´expert. Nous étudions différentes méthodes pour combiner les clusters de mots non supervisés avec les classes sémantiques pour améliorer la reconnaissance d´entités nommées. Les clusters de mots permettent l´obtention de résultats légèrement meilleurs que ceux obtenus au moyen des classes sémantiques à base de connaissance d´expert. Notre étude conclut que les clusters de mots ajoutent une information importante et sont complémentaires des classes sémantiques.

Acknowledgements

First I would like to thank my supervisor Pierre Zweigenbaum who chose me to participate in this study and for his support and helpful guidance. His knowledge and advice have helped me to keep on track and work at a smooth pace. I would also like to thank Cyril Grouin for supporting me throughout this process and for giving me advise. Without their effort, the completion of my master thesis would not have been possible. I also want to thank my other academic supervisor during this Master course, prof. Ioannis G. Tollis.

My gratitude goes out as well in LIMSI team for welcoming me as a member of the group and for helping me complete my internship. My experience over the last six months will be very important for my future and helped me improve my skills.

I wish to thank Lefteris Koumakis and Dimitrios Chasapis for their help and advices. Finally, I want to express my sincere thanks and love to my family and close friends, whose support have given me the motivation to complete this study.

Contents

List of Figures

List of Tables

Introduction

Clinical texts and biomedical literature are important sources of medical knowledge. The exponential growing amount of biomedical sources in recent years has made the Biomedical Text Mining domain highly important. Recent biomedical advances which altered the course of many diseases is based on the understanding of disease mechanisms. Biomedical texts are the most common formal exchange of information, therefore the motivation of trying to extract their information is crucial. Goal of biomedical text mining is to reduce the effort required to obtain useful information from biomedical data by applying automated tools to make this information available to medical professionals and specialized systems. Our methodology contributes to biomedical text mining and especially in named entity recognition by using unsupervised word classes.

Named entity recognition

Named entity recognition is the task that identifies an entity's boundaries within text and assigns the entity to their corresponding class or category. It is usually the first step applied before processing the information contained on biomedical texts . Named entity recognition for biomedical texts is a challenging task due to the dynamic nature of scientific discovery and the amount of semantically relevant entities.

Primary Research Goals

Our primary research goal is to use unsupervised word classes for entity recognition in clinical records. Word classes are set of words that display same properties. They are induced from text corpora and are increasingly used to help tasks that are addressed by supervised classification, such as named entity detection. The principle is based on compensating the lack of coverage of knowledge-based semantic resources by building classes that are adapted to the biomedical domain. Inserting these classes into a supervised learning process leads to the selection of classes subsets and of their words which are relevant for the target task.

We will present named entity recognition systems specialized in the medical domain which learn Conditional Random Fields(CRFs) models [1] from training data based on different attributes. Corpora,which is a large or complete collection of annotated or unannotated texts, used in this research are provided from ShARe/CLEF eHealth Evaluation Lab [2]. A baseline system without semantic classes was initially created [3]. Next steps used preprocessed text that included knowledge-based semantic classes combined with external knowledge sources. The goal of this step was to achieve the best performance of the system based on knowledge-based attributes and it was the starting point of this work.

In our approach we used the following attributes, which are described below:

1. the Brown clusters based on plain text
2. the Brown clusters based on syntactic dependencies
3. the latent categories of the Berkeley parser

To test the hypothesis that syntactic dependencies might improve the task of named entity recognition, we used an often used method for corpus-based word clustering, the Brown clustering algorithm [4]. The Brown clustering algorithm is based on co-occurrence of sequence of two adjacent words in the corpus and creates as output word classes. The Brown algorithm processed the biomedical texts and created attributes based on different variables of the algorithm as additional feature sets on the created CRFs models. The Brown clustering algorithm was also used as an attempt to help supervised learning by [5] and others. The combined method achieved a 25% reduction in errors. Bruijn et al. [6] described three text mining applications and evaluations within the 2010 i2b2 challenge which used the Brown clustering algorithm to create additional feature sets. In any of the previous systems that Brown algorithm was used the input format was string of words. In this work we used the Brown clustering algorithm on syntactic dependencies.

We test if the Brown algorithm by using as input syntactic dependencies could achieve better performance than the Brown clustering method using as input string of words. Different input representations for the syntactic dependencies were created in order to provide the Brown clustering algorithm with the required information. To achieve the best possible performance we experimented with different input representations and different variables of the algorithm. Attributes created were used as additional feature sets for the created CRFs models.

Another approach we tested was whether the latent categories learnt by the Berkeley parser [7] can be used as semantic categories. The Berkeley parser is an implementation of probabilistic parsing with latent categories which produces parse trees based on hierarchical coarse-to-fine parsing and

is considered reasonably efficient. The Berkeley parser was previously used only as a parsing method, so we thought to combine it with supervised machine learning methods to test its ability to improve named entity recognition. The Berkeley parser was used to parse the corpus to create specific syntactic subcategories with the provided English grammar and with a grammar which we created based on the biomedical corpus. Results obtained from the Berkeley parser were also used as additional feature sets for the created CRFs models.

Contents

This thesis is organized as follow. In the first chapter previous literature research related to our work is described. The second chapter provides necessary information about the methods and algorithms we used, details and examples for training and testing data. In third chapter we describe design and implementation of our study including pre-processing steps used for input data, evaluation metrics and detailed steps performed during all the experiments. Final chapter sums up and evaluates our results with final section providing conclusions based on our research.

Chapter 1

Related work

Contents

This chapter surveys previous work in entity recognition. Entity recognition has two approaches : supervised and unsupervised learning methods. Chapter is composed of two sections, supervised named entity recognition and unsupervised word classes, reflecting the goal of this thesis.

1.1 Supervised named entity recognition

Named entity recognition (NER) is a subtask of information extraction that locates boundaries of the entity mentions in a text and classifies them with theirs corresponding semantic types. Biological Named Entity Recognitions automatically identify occurrences of biological or medical terms and assigns them to predefined categories and common entities of interest.

On the other hand, supervised learning is the machine learning task of inferring a function from labelled training data or supervised data. Input-output relationship information is acquired based on a set of paired input-output training sample. In supervised learning, the goal is to predict the

values of the target features for the test examples and unseen examples. Relationship information is frequently represented with learning-model parameters defining the effect of input data on output data. In this section supervised systems are described including a hybrid system , systems based on CRFs, Brown algorithm and Berkeley parser.

1.1.1 Conditional Random Fields

Conditional random fields [1] (CRFs) are a widely used approach in machine learning with many applications in supervised sequence labelling and parsing of sequential data, such as natural language text or biological sequences. Specifically, CRFs can identify applications in shallow parsing, named entity recognition and gene finding based on statistical modelling methods used for structured prediction in pattern recognition. CRFs use a form of discriminative undirected probabilistic graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence. CRFs are an alternative to the related hidden Markov model with the advantage of their conditional nature and their ability to define a much larger set of features [8]. CRFs have the ability to handle large description spaces and to integrate structural dependency between labels [9]. A richer set of label distribution can be modelled because CRFs can use more global features in contrast to HMMs which are local in nature and are constrained to binary transition and emission feature function. Global training makes the training expensive, because it requires a global adjustment of values [10]. Relational data have statistical dependencies that exist between the entities we wish to model and each entity has a rich number of attributes that can improve classification.Graphical models have the ability to exploit the dependence structure among entities but as a result of representing the joint probability of the attributes we wish to predict, graphical modelling can lead to difficulties because it can include very complex dependencies. As a solution it is not possible to ignore this complex models because it will lead in reduced performance but to directly model the conditional distribution.

1.1.2 Natural language processing challenges

Specialized competitions are organized from natural language community (NLP) with aim to improve the state of the art in the domain and identifying diseases in clinical texts. In domain experts organized competitions include Text Retrieval Evaluation Conferences [11], the Semantic Evaluation (SemEval) and the Conference on Natural Language Learning (CoNLL)[12] shared-tasks but results so far showed that research community has still to encounter open research problems in medical domain compared to general domain. Clinical texts are different from other categories of text hence the strategies adopted for name entity recognition in clinical texts are different.

Informatics for Integrating Biology and the Bedside(i2b2) center organized a NLP competitions specialized in medical domain and information extraction from unstructured clinical documents. I2b2 workshops on Natural language Processing attracted international teams and in the third Workshop tackled a set of information extraction problems(drug names and related information, dosage, mode of administration, frequency, duration and reason. Results of third workshop on Natural Language Processing demonstrated that hybrid systems seems very promising as the best performing system was a hybrid [13] although rule-based systems dominated the top 10. Table 7.1 (page 71) shows that hybrid systems performance was significantly different from the other systems and ranked first in both phrase-level and token-level evaluation. The University of Utah, the University of Wisconsin and the University of Sydney used hybrid systems.

The hybrid system of University of Sydney [14] was a complex machine learning model that used Conditional Random Fields (CRF) for named entity recognition and Support Vector Machines for relationship identification. Although performance was optimal compared to other systems, duration and reason were weak . The system used a pre-processing engine for every corpus record before training the CRF to identify the named entity recognition. Preprocessing step included a tokenizer a sentence boundary detector, which relied heavily on a pre-compiled lexicon. After pre-processing completion, seven CRF feature sets were used. For drug entity recognition the feature set included a combination of the results from the drug lexicon verification , drug gazetteer lookup, drug pattern mapping engine and negation engine. Many features were experimented with CRF but the best performance was obtained with this feature set. Results from CRF learner were sent to the SVM feature generator to classify the relationships among entity pairs of medication and its related five entities.

Machine learning approaches for disease mention recognition in biomedical literature were introduced from [15]. Conditional random fields using a set of morphological, orthographic, (i.e. POS, capitalization, digit, punctuation), and shallow syntactic features was implemented aiming in biomedical name entity recognition. The system named BANNER achieved an F-score of 54.84 for disease mention recognition in BioText corpus but BioText corpus contains annotations inconsistencies and it is not suitable for comparing system performance. An improved version of the system achieved an F-score of 77.9 on a different corpus more suitable for disease entity recognition. The AZDC corpus is annotated specifically for benchmarking of disease mention recognition systems. In this system is obvious that the selection of corpus is very important and affects the system performance significantly.

Tang et al. [16] compared performance of CRFs with Structural Support Vector Machines-based NER systems with the same set of features. Data from 2010 i2b2 challenge was used, training set of N = 27,837 and test set N = 45,009. Evaluation showed that structural support vector machines (SSVMs)

improved performance compared to CRFs. However word representations features have shown improvements in performance of the system as in many variety of NLP tasks. Two types of word representations features were used, clustering-based and distributional based features. Both clustering-based and distributional word representations features were benefit in name entity recognition task, but also were complementary to each other. The task was the extraction of clinical entities obtained from clinical narratives, which included discharge summaries and notes. Entities included Problem, Test and Treatment, as training set 349 notes were used and as testing set 477 notes. Both BIO [1]and BIESO [2] format were used and all four categories of features sets including word level information, syntactic information, lexical and semantic information, and discourse information. For word clustering based features Brown algorithm were used and all sub-paths were used as features to represent each word. Both clustering-based and distributional word representations features improved performance in both scheme formats.Table 1.1 provides the results of the best-performed clinical entity recognition systems from both CRFs and SSVMs. Also it shows the entity type used in each system. Although SSVMs achieved higher recall, CRFs achieved higher precision values. SSVMs has strong generalization and therefore has the ability to detect testing samples that do not appear in training data. Word representations mainly improved recall which indicates that more correct entities were detected. System performance increased by each word representation and even more with the combination of both. Therefore more types of word representations or combinations could improve named entity recognition tasks.

Table 1.1 – Counts of different types of entities in training and test data sets used in this study.[16]

Concepts (N = 72.846)				
Data set	**Problem**	**Treat**	**Test**	**All**
Training (349 notes)	11.968	8.500	7.369	27.837
Test (477 notes)	18.550	13.560	12.899	45.009

Petrov et al. [17] continued with implementing a method for pruning in split PCFGS. Presented a comparison of objectives and experiments on automatic splitting for languages other than English. Models were trained for English, German and Chinese, each model was applied directly to tree-banks without any language dependent modification. Parser was outperformed on English by Charniak and Johnson (2005) but performed well across German and Chinese. Parser allows fast, accurate parsing in multiple languages and domains only by using a raw content-free treebank for training and a final

1. B = beginning, I = inside, O = outside
2. B = beginning, I = intermediate, E = end, S = single, O = outside

grammar for decoding, including coarsening maps .

1.1.3 Clustering

The Brown clustering have been used in a variety of NLP application. First attempt to help supervised learning with the use of supervised word classes was the brown clustering combined with discriminative methods by [5]. The combined method achieved a 25% reduction in error on a standard named-entity problem and may also prove useful for low-density languages where limited resources are available. Brown clusters are superior on NER to the word embeddings based on a word representations evaluation [18].Probably because Brown clustering algorithm produces better representations, and most of the errors occurs in rare words. Brown clusters and C&W embeddings have almost the same number of errors which occurred basically in the more common words.If only one word representation is to be used, Brown clusters have the highest accuracy compared to Collobert and Weston [19] embeddings, and HLBL [20].

Combination of unsupervised word classes with supervised learning was used by [21]. Motivation was to incorporate word clusters as additional features for relation extraction. The assumption about unseen words is that other words that share the same cluster may have been seen in the training set. Liang's implementation of Brown clustering algorithm was used with 2 minimum occurrences of words and 1000 clusters. System works by adding an additional layer of lexical features that incorporate words clusters. Semi-supervised system outperformed a state-of-the-art supervised baseline system.

Zhu et al. [22] included brown clustering in a second-ranked model in 2010 I2B2 NLP Challenge. Model aimed to identify semantic relations among medical concepts in clinical text such as problems, tests and treatments. Brown clustering algorithm processed the clinical text and calculated the hierarchical word clusters on the unlabeled data. For each word a unique bit string is assigned and encodes the semantic category of the word. The semantic information of each word is used as a feature in this model and especially the leftmost seven bits of which represent the cluster the word belongs. Although the system ranked second, the results have no significant statistically difference from the rop ranked system and were considerably better than rest systems.

Brujin et al. [6] describes three text mining applications from the National Research Council of Canada on evaluations within the 2010 i2b2 challenge. Each system perform three steps in clinical information extraction. Extraction of medical problems, tests, and treatments from discharge summaries and progress notes; classification of assertions made on the medical problems and classification of relations between medical concepts. All systems were built around a (semi-) supervised machine learning paradigm. Although the learning mechanisms were different for every system, the features sets were mostly similar. Brown clustering algorithm was used as an addition in feature

sets. Brown clustering algorithm clusters form was a mixture of semantic concepts and parts-of-speech. Cross validation at seven hierarchical levels was used to optimize the cluster granularity. First system aim was identification of key concepts anywhere in the source text, including determining the exact boundaries of the concept, as well as the class of the concept. Concepts are non-overlapping and non-nested. System for this task included Brown clustering which generated word-level back off features and amplified in tagging previously unseen words in test data given that those words were seen in the unlabeled data. In table 7.2 (page 71) (c) shows that rich text based features results in performance improvement by 0.09 in recall, 0.022 in precision and 0.020 in F-score compared to order/type sensitive features. These include Brown cluster features besides number of concepts in the sentence, punctuation related features, word n-grams etc.. Syntactic and dependency parsing further improved the results, as did the bootstrapping on the unlabeled data.

Liang [23] used brown algorithm in his master thesis for word clustering and defined the quality of a clustering as the context of a class-based bigram language model 1.1 and the logarithm of probability as 1.2 normalized by the length of the text. Two segmentation tasks that were introduced was named-entity recognition and Chinese word segmentation. For extracting word clusters using Brown clustering algorithm two sets of word clusters were created, one for English and one for German. For both word clusterings 1000 clusters were used. For English word clusters a pre-processing was made by removing any paragraphs that do not resemble real sentences and specifically the ones that are composed of less than 90% lower-case letters. His approach showed that using features obtained from unlabeled data into a supervised model can reduce the errors of the system and the amount of the labelled data needed. Performance of the system can be increased by using word clusters and mutual information statistic but brown clustering result features do not improve the performance in Chinese Words Segmentation. More likely because Chinese characters have more senses than English words and Brown algorithm is a hard word clustering. [23] showed Word clustering features improved performance for name entity recognition in German language as [5] showed that word clustering features improved performance in English language recognition

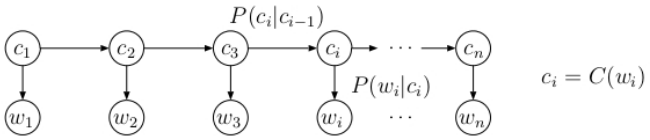

Figure 1.1 – Class-based bigram language model represented as a Bayesion Network, which defines the quality of a clustering. [23]

$$\begin{aligned} \text{Quality}(C) &= \sum_{c,c'} P(c,c') \log \frac{P(c,c')}{P(c)P(c')} + \sum_{w} P(w) \log P(w) \\ &= I(C) - H \end{aligned}$$

Figure 1.2 – Final equation of quality of clustering. I(C) is the mutual information between adjacent clusters and the second term H is the entropy of the word distribution. [23]

1.1.4 Syntactic parsing

Natural language community has been interested in syntactic parsing since it is the first step towards semantic interpretation. Parsers produce a set of possible parses and probabilities for each input sentence, then a second model improves the initial ranking by using additional features of the tree. [24] applied re-ranking to the Berkeley and Brown parsers. Both parsers produce a n-best output, re-ranking was applied to each n-best list of each parser and to the n-best combined list of the union of both parsers. As a straight forward method re-ranking is used for improving accuracy of n-best parsers. Aim was to examine re-ranking with features that improve Brown parser performance can improve also Berkeley parser performance and reverse. Also examine performance of re-ranker trained on the union of n-best output of both parsers. Wall street journal section of the University of Pennsylvania treebank corpus was used, and a 20-fold cross-validation data as described in [25] [26] were used for parsing. Berkeley trainer ran with 6-split and the resulting parser ran in accurate mode and Brown parser in basic settings. Although re-ranking the union of both parsers or the n-best output of Berkeley parser someone would expect to improve the F-score, [24] showed that this is not true. Even though an extended feature set with a wider set of features were used re-ranker performed only marginally better. Probably because the feature of current re-rankers have been designed to perform well with parsers like Brown parser.

1.2 Unsupervised word classes

Contrary to supervised learners, unsupervised learners are not provided with labels. In fact, the basic task of unsupervised learning is to develop the labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group. Word classes are a set of words that display syntax and semantics correlations. They are useful for generalization and abstraction and are used in NLP tasks to add efficiency and increase performance. Supervised systems have the fundamental issue that training data are not capable to provide all the necessary quantitative information for the words that might occur in test data. The potential of word classes obtained through unsupervised machine learning methods is to avoid hand-engineering features and enable computers to automatically process massive amount of unlabelled text. Goal of unsupervised algorithm is to detect similarities in training data and create classes with words based on some correlations. In this section three types of methods that obtain word classes are described.

1.2.1 SEXTANT

A system based on corpus extraction techniques for deriving third-order affinities from a corpus of text have been develop by [27]. Third order techniques create affinities by comparing lists of similar words or groups and terms along semantic axes. SEXTANT system analyses the syntactic usage of word over a corpus and based on this syntactic context calculates similarity of words. Output context of each word is a list of words related by context based on syntactic relation. Problems with this clustering method is that grouping seems to be too fine and in some occasions smaller lists are preferable, noise have to be manually weeded out and words that appear infrequently in the corpus are mostly ignored. Disease entities are not always appear with high frequency in corpus and as a result SEXTANT would ignore many of them.

1.2.2 Brown algorithm

The Brown clustering algorithm [4] is a bottom-up agglomerativeword clustering algorithm for assigning words to classes based on the frequency of their co-occurrence with other words. It is a hierarchical clustering method which optimizes the distribution of words into classes so that the language model learned on bigrams of classes best explains the corpus. Algorithm maximizes the probability of generating the corpus given the language model and generates a hard clustering with each word assigned to one cluster. A binary tree is created as an output from a sequence of words, in which every leaf is a word. A bit of string represents class of the word indicating the path to the root and encoding word's semantic-category information. Words that are similar will

be close to each other in the tree, basically because of the bit that is assigned to each cluster. Intermediate nodes of the tree contain the words in that subtree and in the end all the vocabulary will be a single cluster. To obtain clusters for large vocabularies words are arranged based on their frequency. The Brown clustering algorithm is capable of discovering mistyped words and categorize them to the correct class [3] and has a high degree of capturing syntactic and semantic aspects [4]. In Figure 7.1(page 72) example of subtrees of algorithm are shown.

1.2.3 The Berkeley parser

Parsing is the task of analyzing the grammatical structure of natural language and determining the relations between units based on a sequence of words. The Berkeley parser [7] implements probabilistic parsing with latent categories and combines the strengths of both manual and automatic annotations. It is an implementation of the unlexicalized parsing model with the benefits of being almost language-independent and more generalized. The method produces parse trees based on a hierarchical coarse-to-fine parsing considering a sequence of grammars and language-specific adaptations.

Learning process creates a probabilistic context-free grammar with latent categories. In latent variable parsing rule probabilities learned on latent annotation when marginalized out, maximize likelihood of the unannotated training trees. The method starting with a simple grammar is capable of learning smaller and more accurate grammars than previous grammar refinement work. For rare words a simple and robust method was used by extracting small number of features from the words and then computing approximate tagging probabilities. An unannotated X-bar style grammar is obtained directly from the Tree-bank, tree-bank is a set of sentences annotated with information in form of syntactic parse trees and consist of thousands sentences [28], by the binarization procedure shown in figure 1.3 page 17. The method is capable of learning a probabilistic context-free grammar (PCFG) remarkably good at parsing by beginning with the barest possible initial structure and split-and-merge strategy. Resulting learned grammar is human interpretable although it is entirely automated.

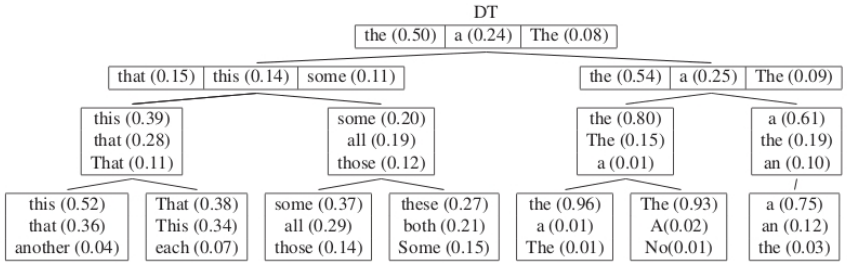

Figure 1.3 – Evolution of the DT tag during hierarchical splitting and merging. Shown are the top three words for each subcategory and their respective probability. [7]

Parsing process produces the most likely parse tree for each sentence based on the created probabilistic context-free grammar and by splitting every non-terminal based on the head word. The Berkeley parser is a well-known PCFG parser which is reasonably efficient and outperformed other parsers on English, German and Chinese at the time it was published. It is an instance of a PCFG-LA parser with the ability to produce latent categories which are finer-grained categories of the original tree-bank grammar on which it is trained. In Table 1.2 (page 17) the rank of method among best lexicalized parsers is presented.

Table 1.2 – Rank results of syntactic parsers. [7]

≤40 words	**LP**	**LR**	**CB**	**0CB**
Klein and Manning(2003)	86.9	85.7	1.10	60.3
Matsuzaki et al. (2005)	86.6	86.7	1.19	61.1
Collins (1999)	88.7	88.5	0.92	66.7
Charniak and Johnson (2005)	90.1	90.1	0.74	70.1
[7]	90.3	90.0	0.78	68.5
all sentences	**LP**	**LR**	**CB**	**0CB**
Klein and Manning(2003)	86.3	85.1	1.31	57.2
Matsuzaki et al. (2005)	86.1	86.0	1.39	58.3
Collins (1999)	88.3	88.1	1.06	64.0
Charniak and Johnson (2005)	89.5	89.6	0.88	67.6
[7]	89.8	89.6	0.92	66.3

Chapter 2

Materials and methods

Contents

This section provides an in-depth discussion of materials used in this study. Two algorithm implementations were used (a) the Brown clustering algorithm and (b) the Berkeley Parser, conditional random fields implemented by a toolkit and input data of biological genre. First section presents the input data used for training and testing, their preprocessing and categories. Chapter continues with description of CRFs definition and input formats, tag for entities and Wapiti toolkit description. Finally data format of the Brown algorithm and the Berkeley parser are described, variables used and basic steps of each algorithm.

2.1 Corpora

Corpora is a large or complete collection of annotated or unannotated texts, providing lexical information, semantic information and morphosyntactic information in linguistics analysis. Unannotated corpora are raw states of plain text with considerably use in natural language processing field because of their large number. Annotated corpora are generally small in size and may

be manually annotated by humans or automatically, semi-automatically algorithms whose outputs have to be post-processed by humans. Annotation provides easier retrieval and analysis of information of the texts contained in the corpus.

2.1.1 Annotated corpus for medical entity detection

Although annotated corpora is not required, often are more useful than raw text. There are many resources for text mining: MEDLINE that contains references in a variety of medical texts, like journal articles in the life sciences that are maintained from National Library of Medicine(NLM), topically annotated collections like BioCreAtIve collections and PennVioIE corpus, individual research groups annotations and finally Collaborative Annotation of a Large Biomedical Corpus (CALBC) that proposes the use of a silver standard corpus with annotated data that have been produced automatically. Several recent researches have used the Web as a corpus and several collections of clinical text have become public the recent years. Some of this collections are the Pittsburg collection of clinical reports [29] , the annotated i2b2 collections [30, 13] and reports in the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC II) [31]

This work uses the training corpus provided from ShARe/CLEF eHealth Evaluation Lab [2] which contains 200 clinical reports with stand-off annotations of disorder mention spans and UMLS concept unique identifiers (CUIs) with a total of 94,243 words. Dataset consists of de-identified clinical free-text notes from MIMIC II database version 2.5(mimic.physionet.org) [31]. MIMIC II database is composed of two distinctive data groups. The first group contains patient demographics, medications, results of lab tests and more, integrated from different information systems. The second group contains high resolution waveforms recorded from monitors in care units. In this challenge notes are in the ICU setting and included types are ECG reports, echography reports and radiology reports. It is generally recorded manually and requires infrequent updates. Admission/discharge only occurs once during a patient stay. ICU transfer will only occur a few times. Medication will only occur a few times a day and reports are added when particular diagnostics are performed. All the above information can be considered as discrete patient data.

The task consisted in automatically identifying the boundaries of disorder entities in the text and to map them, as a separate subtask , to SNOMED codes . Annotation of disorders was accomplished by two professional coders trained for this task and followed by an open adjudication step. A concept is part of the disorder semantic group if it belongs to one of the following UMLS semantic types:

- Congenital Abnormality

- Acquired Abnormality

- Injury or Poisoning
- Pathologic Function
- Disease or Syndrome
- Mental or Behavioral Dysfunction
- Cell or Molecular Dysfunction
- Experimental Model of Disease
- Anatomical Abnormality
- Neoplastic Process
- Signs and Symptoms

Annotations are stand-off and are in the following format

synthetic example: report name || annotation type || cui || char start || char end

Some examples of the corpus:

26563-387055-RADIOLOGY_REPORT.txt||Disease_Disorder||C0032326||674||686
26563-387055-RADIOLOGY_REPORT.txt||Disease_Disorder||CUI-less||977||982||993||1001
26563-387055-RADIOLOGY_REPORT.txt||Disease_Disorder||CUI-less||1060||1093
26563-387055-RADIOLOGY_REPORT.txt||Disease_Disorder||C0032227||1154||1170

In first example, disorder is token "pneumothorax", next is the cui of the token "C0032326" and the spans of the word. In second example we have a discontinuous entity, which are very interesting because their processing is very difficult. Sentence in the file is "heart is mildly enlarged" but disorder entity is the "heart enlarged" , this is why number of spans are discontinuous, also the CUI value “CUI-less” is the value assigned when there is no CUI found in the database. Next example is the token "pulmonary vascular redistribution" which is a continuous disorder entity, CUI is also not found for the entity. Last example is "pleural effusion" which is a continuous disorder entity with an existing CUI, “C0032227”.

2.1.2 Unannotated corpus for semantic class induction

Unannotated corpora are larger than annotated corpora rendering them useful for unsupervised methods to learn word classes and help entity detection in the annotated corpus. Important point is that they have to be of similar genre and style with the domain intended to be used. Methods of the present work aim to learn word classes from unannotated corpora to improve supervised learning from annotated corpus by using part of the unannotated

MIMIC II Database data and in particular the discharge summaries. The corpus we used contains 18338 discharge summaries, from which those in the 200 training set were removed. Discharge summaries are free-text fields that nurses can enter any information and present a summary of the entire hospitalization period of the patient.

2.2 Conditional Random Fields

2.2.1 Defintion

A conditional random field [1] (CRFs) is a conditional distribution with an associated graphical structure. Dependencies among input variables do not need to be explicitly represented and as a result it is possible to have as input large number of global attributes, such as combination of neigh-boring words, bigrams, Part-of-speech , prefixes , suffixes and many others. In other words, if we wish to predict a large number of variables that have dependencies with each other CRFs are essential because they combine the advantages of classification methods to predict using large amount of input attributes and the advantages of graphical modelling to model multivariate data. Input provided to CRFs are multiple tokens with each token represented by a vector of attributes. Definition of token depends on tasks, generally is a characters string between two spaces, depending on the tokenization made, a token can include a word and the surrounding punctuation mark. Conditional Random Fields must be provided with patterns which specify how feature functions will be calculated, i.e. which attributes will be used in training and testing and their combinations.

2.2.2 Tag for entities

Appropriate tag representations should be assigned in an NER task to be transformed in a classification problem. The BIO tagging scheme [32] is a commonly used representation of entity tags, where individual tags are assigned to each word as following: B = beginning of an entity, I = inside of an entity and O = outside of an entity. Beside the BIO tagging scheme there is another type of tag representation called BIESO, in which each word is classified as: B = beginning, I = intermediate, E = end, S = single word entity and O = outside. BIO format have been used with great success in NLP tasks but it can be problematic if entity boundaries overlap and the problem of recognizing nested biomedical entities has been addressed by [33] [34]. Zeng et al. [35] showed that BIESO tag had better performance in clinical entity recognition, therefore sometimes both tag representations are included. It is not clear which type of tagging scheme performs better, generally performance of entity tag representation is affected of the task and the variables of the current system.

2.3 WAPITI tool-kit

Wapiti [9] is a simple and fast discriminative sequence labelling tool-kit for segmenting and labeling discriminative models. It is based on maxent models, maximum entropy Markov models and linear-chain CRF and proposes various optimization and regularization methods to improve both the computational complexity and the prediction performance of standard models. Wapiti is developed by LIMSI-CNRS and was partially funded by ANR projects Crotal (ANR-07-MDCO-003) and MGA (ANR-07-BLAN-0311-02).

Wapiti toolkit is an implementation of Conditional Random Fields. We describe the format of the training and test files. Input datasets must be in a tabular form with each column, separated by spaces or tabulations, representing an attribute and each row representing a vector of attributes of each token. Tabular form must have a fixed number of columns and be consisted of multiple tokens. To identify the boundary between sentences an empty line is added. All tokens are observations available for training or labeling, except the last one in training mode which is assumed to be the label to predict. Representation of entities tags of tokens are in BIO format which is described in the previous chapter. Patterns are given as input to specify in advance how feature functions will be computed. In pattern files each line is a pattern, empty lines and characters appearing after '#' are discarded. Special macro %x[row,col] is used to specify a token, row specifies the relative position from the current focusing token and col specifies the column of the attribute. There are three types of templates : Unigram defined with u, bigrams defined with b and both defined with '*'. The actual features used by the CRF are computed based on this patterns and the possible values of each special macro. Features used by the Wapiti can be displayed with their probabilities by dumping the model into a text file, as shown in the example for attribute lemmatization in position 0 and token value 'allergy':

*:LEMMA:1:+0:+0:allergy	#	O	0.257384
*:LEMMA:1:+0:+0:allergy	#	B-DISORDER	-0.257384
*:LEMMA:1:+0:+0:allergy	#	I-DISORDER	-0.094349
*:LEMMA:1:+0:+0:allergy	O	O	0.094349
*:LEMMA:1:+0:+0:allergy	O	B-DISORDER	-0.094349
*:LEMMA:1:+0:+0:allergy	B-DISORDER	I-DISORDER	-0.094349

Output files have the same format as input files with an extra column added in the end displaying the predicted class of each token. Predicted tags are in the same scheme format as input tags.

2.4 Brown clustering algorithm

Percy Liang's [23] implementation is used in this study. Input of the algorithm is sequence of words of raw text separated by whitespace. Each sentence of the text must be presented in separate line. Suppose that a partition of a vocabulary of V words into C classes is created. Brown clustering algorithm creates C distinct classes based on the top C most frequent words in the corpus. At the first step the $(\mathbf{C}+\mathbf{1})^{\mathbf{st}}$ most probable word is assigned to a new class and then the pair of classes for which the loss in average mutual information is least are merged. After V-C merges, C classes remain and each word in the vocabulary has been assigned to a class. The average mutual information can be increased by moving some words from one class to another. Brown clustering algorithm cycles through the classes and moves words through classes aiming at a partition with a better average mutual information until no reassignment can be done. A better partition is possible to be found by moving two or more words through classes but it would be very costly. In Algorithm 1 a pseudo code of Brown clustering algorithm is described.

Algorithm 1 Brown clustering algorithm

Require: $m = 1000$
1: Take top m most frequent words
2: Put each word in a cluster
3: **for** $i = (m+1) \rightarrow V$ **do**
4: Create a new cluster $c_{\text{m+1}}$ for the i_{th} most frequent word.
5: Choose two clusters from $c_1 \ldots c_{\text{m+1}}$ to be merged
6: Compute the Average Mutual Information
7: **end for**
8: Select pair of cluster with least loss of Average Mutual Information
9: Carry out $(m-1)$ final merges, to create a full hierarchy

Output of the algorithm is a file contains for each token a cluster and in particular in each line : ⟨bit string of cluster⟩ ⟨token⟩ ⟨number of word's occurrences in input⟩

101111000	Bupropion	2
101111000	Malignant	2
101111000	Hepatitis-C	2
01111000111	Olanzapine	2
101111000	anicteric	2

In figure 2.1 (page 25) examples of Brown clusters on MIMIC II corpus are shown.

ECG_REPORT DISCHARGE_SUMMARY RADIOLOGY_REPORT ECHO_REPORT

surgeries films recommendation angiograms hematoma results recommendations findings images

dilaudid prescription plan candidate month, Lipitor glass trial half month week unit

liters. subcutaneously. prophylaxis. children. years, months. years. weeks.

Bupropion Malignant Hepatitis-C Zidovudine Phenytoin Baclofen Sevelamer Thiamine Gastroesophageal Renal Schizoaffective PULMONARY

multivitamin Ezetimibe Nystatin Zoloft Desipramine Ulcerative Polyvinyl Hydralazine Magnesium loratadine Lamivudine Oxycodone

fever. today. leukocytosis. agitation. details. that. head. pain. evaluation.

[**2016-03-14**] [**2013-06-04**] [**2020-01-20**] [**2015-07-19**] [**2014-07-13**] [**2019-03-19**] [**2014-01-26**]

Figure 2.1 – Example of clusters.

2.5 The Berkeley parser

The Berkeley parser uses as input a grammar to assign the most likely parse tree of each sentence. Provided languages of the parser are English, German, French, Bulgarian, Arabic and Chinese. Input data of the algorithm are tokenized sentences with one sentence per line. Output of the parser is a parse tree in Penn TreeBank format shown in figure 2.3 (page 27). In algorithm 2 (page 25) a pseudo code of Berkeley parser is presented.

Algorithm 2 Berkeley parser

1: Begin with a string consisting of the start symbol
2: **repeat**
3: split every non-terminal X in the string into two non-terminals
4: $X \rightarrow Y_1 \ldots Y_n$
5: learn PCFG through EM {#because the newly split non-terminals are not exactly observed in the dataset}
6: for each split, compute the likelihood loss when the split is undone
7: **if** loss is too little **then**
8: Undo the splitting
9: **end if**
10: **until** until there are only terminals in the string

Output parse trees can be given to the TreeLabeler, a tool for annotation of parse trees with their most likely Viterbi derivation over refined categories. Output of the tool are latent categories, an example of them is provided in figure 2.2, based on corpora described and parsed with English grammar. As shown in the figure subcategories of the tag sets are added, refining base treebank symbols to improve statistical fit of the grammar.

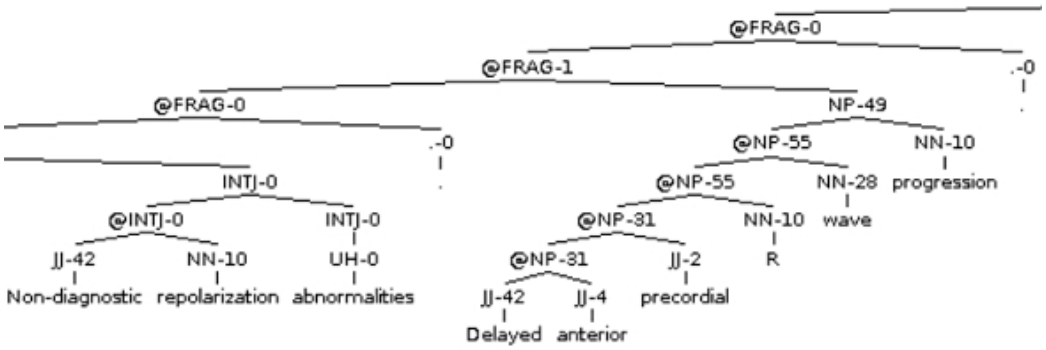

Figure 2.2 – Subtree of annotated parse tree. [36]

For training a grammar, a treebank is needed in Penn TreeBank format. The Berkeley parser splits, merges and smooths the grammar by creating an intermediate grammar file once in a while. The process is expected to complete after several hours or days based on the input treebank. Parser also provides testing of the performance of the grammar and the option to export the grammar in text format for examination.

The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	'	Left open single quote
22.	RBS	Adverb, superlative	46.	"	Left open double quote
23.	RP	Particle	47.	'	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	"	Right close double quote

Figure 2.3 – The Penn TreeBank POS tagset. [36]

Chapter 3

Design and Implementation

Contents

In this chapter we describe the task of disorder detection in clinical reports in English language. We also discuss the challenges encountered and efforts which could maximize the performance of the algorithm. Wapiti tool which implements Conditional Random Fields to create a supervised linear-chain model and datasets described in Chapter Materials and methods in section 2.1 (page 19) was used. Chapter describes steps of supervised entity recognition in this study and continues with the preprocessing of corpora used. Based on the goals of the overall study knowledge-based semantic classes and automatically induced classes, including Brown clusters computed on string of words and on syntactic dependencies and latent categories of the Berkeley parser, are described. Final section is design of the experiments for each of the above described classes.

3.1 Supervised entity detection

Wapiti toolkit implements conditional Random fields, as described in section 2.2 (page22) input should be in tabular form which represents each word as an attribute vector. Each line represents word as an attribute vector and each column represents one attribute. Second column is token of the word and following columns are the attributes we have included. Last column contains the true class of each token. In this work the chosen representation was each line to begin with the filename were word was detected and the start and end of word's character in the file. Attributes added are the semantic classes presented in next chapters. Patterns defining the actual features used by the CRF were created including unigrams, bigrams and both also and combinations of bigrams and unigrams of attributes.

3.1.1 Evaluation metrics

System's ability to correctly identify mentions of entity noun phrases is evaluated with three measures: precision, recall and F-measure

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

$$\mathbf{F - measure} = \frac{\mathbf{2 * Precision * Recall}}{\mathbf{Precision + Recall}}$$

TP = count of system Annotation of entity noun phrases presenting same span as gold standard NPs

FP = count of system Annotation of entity noun phrases presenting divergent span as gold standard NPs

FN = count of gold standard entity noun phrases not present in the system disorder entities

Precision or Confidence denotes the proportion of Predicted Positive cases that are correctly Real Positives. Recall or Sensitivity is the proportion of Real Positive cases that are correctly Predicted Positive. F-measure is a measure of experiment's accuracy. It considers both precision and recall of the test to compute a harmonic average. It references the True Positives to the Arithmetic Mean of Predicted Positives and Real Positives, being a constructed rate normalized to an idealized value [37]. For classification tasks terms positive and negative refer to the classifier's prediction and the terms true and false refer to whether that prediction corresponds to the true class. This is shown by the table below:

		Actual class (observation)
predicted class (expectation)	tp (true positive) Correct result	fp (false positive) Unexpected result
	fn (false negative) Missing result	tn (true negative) Correct absence of result

The goal of entity detection is to optimize F-measure. Important point is the difference between token and entity. Entity could be more than one word including a begin word and an end word and in many cases intermediate words.

3.1.2 Performance estimation

For each produced model performance estimation is required to select best model out of all possible models. Ideal performance estimation would be to learn a model from samples in train set, observe its operation for some time in test cases and estimate the performance of the model on testing examples. To simulate the best possible performance estimation a 10-fold cross validation was used. Data was randomly split in 10-folds of equal size and in each iteration nine folds were used as training samples and one fold as test sample resulting in a model for each iteration. Figure 3.1 (page 31) provides a visualization of the 10-fold cross validation.

Figure 3.1 – 10 folds of cross validation

.

Final performance was obtained by calculating the average of the 10 iteration based on a loss function which measures the discrepancy between truth and prediction.

$$performanceestimation : \frac{1}{10}\sum_{i=1}^{10} f(Train, Test_i)$$

where f(Train,Test$_i$) is the ability of the model to correctly identify spans of disorder noun phrases.

3.2 Data pre-processing

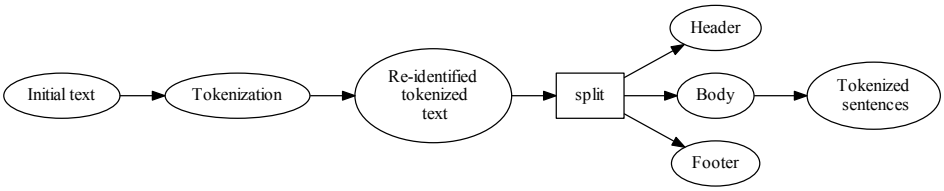

Figure 3.2 – Pre-processing steps

.

This work reuses a pipeline of components prepared for the CLEF eHealth challenge (Bodnari et al.) [3]. The features produced by this pipeline included knowledge-based semantic classes, but no data-driven classes. We have added the latter and performed experiments to test their contribution. Several pre-processing steps were performed before the use of corpora. Training and test corpora contained special de-identification marks and documents form consisted of header, body document and footer. Disorders information was only present in the body of the document with header and footer containing information about clinical administration. Re-identification with pseudonyms were performed to create more normal phrases. Header and footer was removed and only body of documents was analyzed. Steps of pre-processing described above are provided in Figure 3.2.

3.3 System features

Features of the baseline system was lexical and morphological features, syntactic features and document structure features. Lexical and morphological features included token, token lemma, characteristics such as token containing only upper case letters, token is a digit, is capitalized and is a punctuation. Syntactic features contained part of speech information which extracted by using cTakes [38] system on the input text. Document structure features included information about the type of the document and the section type which was extracted with a rule-based section extraction tool that identifies the occurrence of section names within the text.

3.4 Knowledge-based semantic classes

Disease entity recognition is a subtask of information extraction targeting to locate the disease mentions in text. Semantics in linguistics is the sub-

field that is devoted to the study of meaning. Although conditional random fields and especially tool Wapiti has given promising results in disorder detection from clinical texts, features based on semantic categories were added to explore the possibility of achieving better results by adding the additional information in our system. Knowledge-based semantic classes were included in the system prepared by Bodnari et al. [3]. In figure 3.3 are shown the steps that will be described in the next sections.

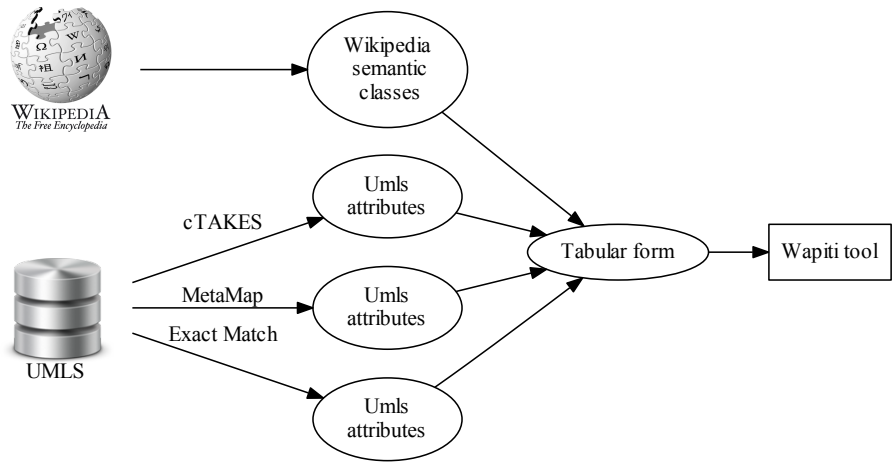

Figure 3.3 – Adding semantic resources information in the initial system.

3.4.1 Unified Medical Language System(UMLS) categories

The UMLS [39] is a resource for biomedical systems and services which has the ability to link health information, medical terms, drug names and billing codes across different computer systems. The UMLS has three knowledge sources:

- Metathesaurus: Terms and codes from many vocabularies
- Semantic Network: Broad categories (semantic types) and their relationships (semantic relations)
- SPECIALIST Lexicon and Lexical Tools: Natural language processing tools

UMLS features included :

- Semantic group information

- Unique identifiers for concept (basic unit in Metathesaurus)

- Unique identifier for semantic types(broad categories which is assigned to each Metathesaurus concept) provided by cTakes which contains anatomical sites, procedures, signs/symptoms.

- Identified group category resulting by processing the input text with MetaMap[40].

- Semantic group

- Concept unique identifier of exact match with UMLS noun phrase

- Medication

- Measurement

- Two features of binary value were also used indicating the exact match with the disorder semantic group and with anatomy semantic group

Corpus were processed and UMLS sources were applied resulting in distinct values of each UMLS source:

Attribute	**Distinct value**
Feature dependency semantic group	156
Concept unique identifier	3308
Semantic type	128
Semantic group	26
UMLS category	18
Lexicon concept unique identifier	4204
Lexicon semantic group	18
Medication	3
Measurement	3

Some of the most informative UMLS features based on the experiments are presented in Table 3.1 (page 35).

Table 3.1 – Umls attributes for some tokens.

Token	**Dependency semantic group**	**CUI**	**TUI**	**Semantic group**	**UMLS category**	**Lexicon semantic category**	**Lexicon CUI**	**Class**
Drugs	PMOD_DISO	I-C0013182	I-T047	I-DISO	O	B-CHEM	B-C0013227	O
altered	NMOD_FIND	#NA	#NA	#NA	B_DISO	O	O	B-DISORDER
mental	NMOD_FIND	B-C0278060	B-T033	B-FIND	I_DISO	B-PHYS	B-C0229992	I-DISORDER
status	ROOT	I-C0278060	I-T033	I-FIND	I_DISO	O	O	I-DISORDER
He	SBJ_#NA	#NA	#NA	#NA	O	O	O	O
also	ADV_#NA	#NA	#NA	#NA	O	O	O	O
received	ROOT	#NA	#NA	#NA	O	O	O	O
levofloxacin	OBJ_#NA	#NA	#NA	#NA	O	B-PHYS	B-C0803434	O
and	COORD_#NA	#NA	#NA	#NA	O	O	O	O
flagyl	CONJ_#NA	#NA	#NA	#NA	B_DISO	B-CHEM	B-C0699678	O
for	NMOD_#NA	#NA	#NA	#NA	O	O	O	O
aspiration	NMOD_DISO	B-C0032290	B-T047	B-DISO	B_DISO	B-DISO	B-C0032290	B-DISORDER
pneumonia	PMOD_#NA	I-C0032290	I-T047	I-DISO	I_DISO	I-DISO	I-C0032290	I-DISORDER
.	P_#NA	#NA	#NA	#NA	O	O	O	O

In table 3.1 there is an example of a single word, a disorder, and a sentence containing a disorder. First column separated with a double vertical line contains the tokens of each example, last column contains the correct class of each token, B-DISORDER for begin of disorder entity, I-DISORDER for intermediate token of entity and O for outside entity, and rest of the columns contain the UMLS attribute's values. Each example is separated with a double horizontal line. As it is shown there are some errors in the categories like the "levofloxacin" categorized as B-PHYS, which is a category containing tokens like "mental, history, hospital, male", instead of B-CHEM . UMLS categories B-DISO, I-DISO in this example are correctly assigned corresponding to classes, except token "flagyl" which is categorized as a B-DISO class instead of O class. Concept unique identifier (CUI), Semantic group type (TUI) and semantic group in non disorder tokens contain the value #NA indicating that it is not a disorder.

Table 3.2 (page 36) provides examples of the semantic type. The first semantic type "B-T019" contains words that in their majority are a disorder or the beginning of a disorder, on the contrary the second semantic group type contains words that are disorders but also they are not part of disorder entities. This example and previous errors in table of UMLS features showed us that a single feature is not enough for a correct classification and combination of features is required.

Table 3.2 – Semantic types unique identifier

Semantic type name	**TUI**	**Words**	**Class**
Congenital Abnormality	B-T019	atrial	B-DISORDER
		bicuspid	B-DISORDER
		Ebsteins	B-DISORDER
		myelocele	B-DISORDER
		retractile	B-DISORDER
		multiple	O
Injury of poisoning	B-T037	abrasion	B-DISORDER & O
		bone	I-DISORDER
		colonic	I-DISORDER
		dislocation	B-DISORDER & O
		exposure	B-DISORDER & O
		femoral	B-DISORDER

3.4.2 Wikipedia categories

Semantic resources has the disadvantage of lacking precision since they are not adapted in a specific domain or type of texts. Wikipedia categories is an additional attempt to add information in the system. Wikipedia was used to create two new attributes which included semantic groups of Wikipedia and disease and body parts concepts. Wikipedia categories are grouped in nine groups: disorder, body part, living being, chemicals, phenomenon, object, geographical location, devices and other. In table 3.3 (page 36) are shown the two features of Wikipedia, Wikipedia semantic group and Wikipedia disease and body parts.

Table 3.3 – Semantic types unique identifier

Token	**Wikipedia semantic groups**	**Wikipedia disease & body parts**	**Class**
adrenal	BBODYPART	BBODYPART	O
Advair	BCHEMICALS	#NA	O
adventitious	BLIVEBEING	#NA	B-DISORDER
Amylase	BCHEMICALS	#NA	O
anteverted	BBODYPART	BBODYPART	B-DISORDER & O
awareness	BPHYS	#NA	I-DISORDER
bed	IDEVICES	#NA	O
life	BPHENOMEN	#NA	O
adult	BDISEASE	BDISEASE	B-DISORDER

Attributes can be complementary by giving to the system more information for a token if it is a body part or a disease since these two categories have equal values in both features. Wikipedia semantic groups attributes have more categories like living being, devices, chemicals, phenomenon etc.

3.5 Data-driven word classes

Word classes extracted from text corpora, typically through distributional analysis, are increasingly used to help tasks that are addressed by supervised classification such as named entity detection. As mentioned in previous chapter semantic resources sometimes lack of precision and often lack of coverage, the principle consists of compensating this disadvantage by building classes that are adapted to a domain and to a genre of text. Contribution of the present work is the computation and addition of data-driven word classes based on two different ways of applying the Brown clustering algorithm and on the latent categories of the Berkeley parser. More specifically, inserting these classes into a supervised learning process leads to the selection of subsets of these classes and of their words which are relevant for the target task.

3.5.1 Brown clustering algorithm

The Brown clustering algorithm presented in section 2.4 (page 24) has been used in unannotated corpora described in section 2.1.2 in order to classify the entities and add information to the system. Different numbers of clusters and minimum occurrences for each word to be classified have been tested. Initially the Brown clustering algorithm was used with flat files resulting from a corpus containing 18338 discharge summaries. Figure presents the steps and two different approaches, Brown clustering with conventional input and our approach to improve named entity recognition in this study.

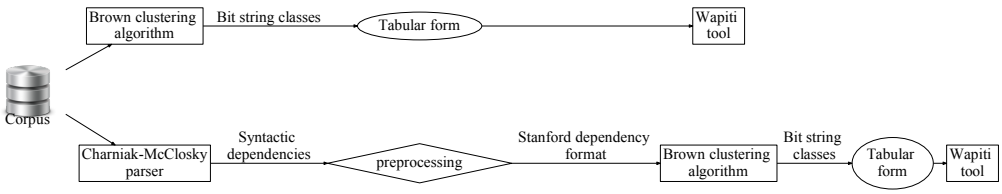

Figure 3.4 – Adding semantic resources information in the initial system.

Our approach was to experiment with the Brown clustering algorithm by providing as input different data to find whether they can fare better than the algorithm with input string of words. We test the hypothesis that syntactic dependencies might be more appropriate to compute these clusters. Charniak-McClosky parser [41] was used in the same corpus to obtain the syntactic dependencies and the output files were converted into Stanford dependency format [42]. Stanford dependency format maps straightforwardly onto a directed graph representation, in which words are nodes and grammatical relations are edges.

The Brown algorithm has been designed to process words as input, and in this work an attempt to use it with syntactic dependencies is made . Changes

have not been applied in the Brown algorithm but instead we tried to find a representation of the syntactic dependencies that would be processed directly by the algorithm and the implementation that we use, i.e Percy Liang's code [23]. Brown algorithm assigns each word to a class but is based on the co-occurrences of words as bigrams. Since the Brown clustering algorithm is designed to use words as input, the transformation to tokens with their syntactic dependencies as input is a complicated task in order to add the information correctly to the algorithm.

Files were preprocessed by removing all unnecessary symbols and keeping only two tokens with their relation per sentence with first token to be the governor and the next token to be the dependent. Initially four types of files were created with the following input representations :

- token-relation-token
- relation-token-token
- token-token-relation
- relation-token-token-relation

The last input representation (relation-token-token-relation) will not be further examined because results didn't improve performance. In Table 3.4 examples of the first three formats are provided.

Table 3.4 – Examples of syntactic dependencies representations

Input representation	**Example**
token-relation-token	atrium det The atrium amod left dilated nsubjpass atrium dilated auxpass is dilated advmod mildly ROOT root dilated
relation-token-token	det atrium The amod atrium left nsubjpass dilated atrium auxpass dilated is advmod dilated mildly root ROOT dilated
token-token-relation	atrium The det atrium left amod dilated atrium nsubjpass dilated is auxpass dilated mildly advmod ROOT dilated root

Second approach was to transform relation and tokens in sentences of two words. Several combinations were used :

- relation conjoined with the first token
- second token conjoined with the relation

- two tokens without the relation

The above types were created in order to input correctly the information in the algorithm since the algorithm would process the input strings as bigrams meaning that in the previous approach it would examine token with another token or token with the syntactic dependency and each time the information would not include the two tokens and their relation. In Table 3.4 examples of the first three formats are provided.

Table 3.5 – Examples of syntactic dependencies representations

Input representation	**Example**
first token conjoined with the relation	det_atrium The amod_atrium left conj_2017-07-06 atrium auxpass_dilated is advmod_dilated mildly root_ROOT dilated
second token conjoined with the relation	atrium det_The atrium amod_left 2017-07-06 conj_atrium dilated auxpass_is dilated advmod_mildly ROOT root_dilated
two tokens without the relation	atrium The atrium left dilated atrium dilated is dilated mildly ROOT dilated

Resulting files were given as input in Brown clustering algorithm. Minimum occurrences used were 2, 4, 8, 16, 32 and number of clusters 32, 100, 320, 1000 in various combinations. A lower threshold for number of occurrences includes a larger number of words in the resulting lexicon, hence is expected to better cover the vocabulary of the corpus but a very small threshold may lead to less accurate clustering because of the low-occurring words. Resulting to a trade-off between recall and precision. Numbers of clusters tested grow with $\mathbf{sqrt(10)}$ is because that multiplies by 10 the processing time, complexity of the Brown clustering algorithm is $\mathbf{O(nc^2)}$ where n is the numbers of words and c is the number of clusters.

In Table 3.6 some examples of the input representations are provided. Extended examples are provided in Table 8.1 (page 76). First column of the table contains the input representation, the third column contains tokens in-

cluded in Brown clustering classes, tokens “stroke” and “hemiparesis” were selected as examples, and the second column contains the classes assigned by Brown clustering algorithm based on the second approach to these tokens. First line contains the classes assigned by Brown algorithm with input string of words and rest of the multiple-lines contain classes assigned by the algorithm with the input files containing syntactic dependencies. In some examples token “stroke” and “hemiparesis” are observed as single words and in others they are part of the conjoined relation-token entity, based on their position in the sentence as governor or as dependent .

Input representation	Class	Tokens included in class
original text files	11111111110011111	nurse ileostomy wh PEG face pancreatitis pacemaker SBP GI drain **stroke**
	111111101111011	disc main anterior **hemiparesis**
token relation token	11111101111111	endocarditis dementia rectum radiation trauma sepsis seizure **stroke** bleeding fluid mass
	111111001001	aphasia tachypnea hemorrhoids toxicity ketoacidosis hyponatremia hypothyroidism nephropathy obesity **hemiparesis** flare thrombocytopenia
relation-token token	011110001	syndrome ischemia hernia stroke dysfunction bruits lymphadenopathy **stroke** erythema Diabetes
	1111100100	prep_from_shifts **conj_vs_stroke** prep_of_staging conj_but_anemia prep_due_to_liver conj_or_hypercholesterolemia
token token - relation	011101001101	inflammation necrosis renal involvement pathology acute compression trauma **stroke** ischemia abscess infarct hemorrhage cephaloid
	101111011100	root_palpable root_DVT root_distention root_crackles root_ventricle root_intervention **root_stroke** root_membranes root_ischemia root_region
token token	11101100	disorder Anemia sepsis seizure PE ischemia **stroke** Disease rash aspiration ucler hypotension GI anemia
	110111010	cholecystitis appendage complex deformity perforation hydrocephalus **hemiparesis** infarcts emphysema necrosis views dilation

Table 3.6 – Examples of Brown clusters obtained based on representation of syntactic dependencies.

3.5.2 Berkeley parser

Latent categories learned by the Berkeley parser presented in Chapter 2.5 (page25) have been extracted and used as semantic categories to test the hypothesis and check if can increase the performance of the system. The same corpus, 18338 discharge summaries, used in Brown algorithm have been also used in Berkeley parser. Figure 3.5 are provides the steps described in this section.

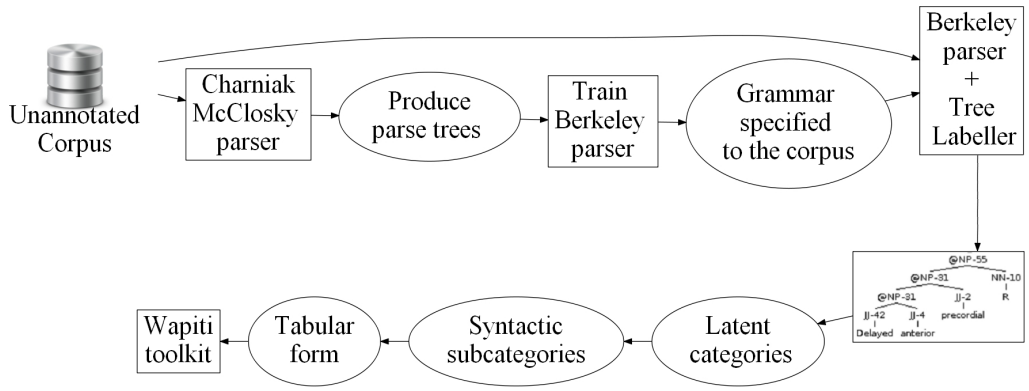

Figure 3.5 – Adding semantic resources information in the initial system.

First approach was parsing of the corpus and collecting the output trees with the semantic categories based on English grammar provided by the Berkeley parser. Output trees were used as input in Berkeley tree labeller with objective to use the output subcategories as semantic categories. Output trees were processed and intermediate latent categories for each word of the input sentences were extracted. For each word in the corpus possible subcategories were collected including their frequency on the word. Each word had many different subcategories therefore we decided to collect the three subcategories with higher frequency to create three lexicons. Each lexicon was used to create an additional attribute for each token, added on the initial system.

The default English grammar of the Berkeley parser is not based on medical texts, so we expect that a better performance in disease entity recognition can be achieved by using a grammar adapted to the specific domain and genre of texts. Second approach was to create a grammar based on the specific type of texts. Ten percent of the corpus was parsed by the Charniak-McClosky parser [41], which itself is self-trained on biomedical texts, to obtain a treebank in which a new grammar was based. That parsed corpus was used to train the grammar consisting of 1841 files, 356867 sentences for training and

196 files, 35931 sentences for tuning. The resulting grammar was adapted on this corpus and was used by the Berkeley parser to parse the larger corpus. The output trees of the new parsing were further processed with the Berkeley tree labeller and subcategories were collected with the same method as previous approach. Three lexicons with the most frequent subcategories for each word were created and the resulting attributes were added to the initial system.

In Table 3.7 examples of subcategories are provided. Table 3.7 is divided in the first two columns referred to default grammar provided by the parser and to the last two columns referred to our grammar. We have chosen two words "stroke" and "hemiparesis" which are disorders or part of entity of disorders. Based on each of the two words the three top subcategories were extracted. In first and third column, these top subcategories are provided and more precisely first three rows are referred to "stroke" and last three rows to "hemiparesis". We provide only the three top subcategories because as mentioned above there are many subcategories for each word and we decided to choose the three most frequent.

Table 3.7 – Berkeley parser token's three most frequent subcategories.

Top three subcategories			
Default Grammar		**Our grammar**	
Category	**Tokens in category**	**Category**	**Tokens in category**
NN-10	**stroke** studding telithromycin tenderness unsteadiness ureter accreta actonoel	NN-18	**stroke** tenderness adenocarcinoma supplementation cancer symmetry Colectomy syndrome tachycardia territory suppression
NN-49	**stroke** studding surgery topiramate vasospasm acuity alcoholismcaspofungin	NN-30	**stroke** callosum subfalcine suctioning nursery suicide pneumonectomy aspect supplementation suppression susceptibility
NN-31	**stroke** triangular ventricularn amuptation blister certification	JJ-1	**stroke** subcarinal thrombotic tunneled undetectable vocal average cerebrospinal contralateral endometrial
NN-10	**hemiparesis** hemiplegia immobilization infection keratosis Knapton aortography	NN-19	**hemiparesis** hemiplegia hemisphere hemithorax hepatis hilum ileum ilium imaging immunization
NN-31	**hemiparesis** inrtaop kinase lymphocyte nephrectomy nocardia paucity perineum	NN-18	**hemiparesis** hemiplegia hemorrhoid insuffiency intervention hydrocele hydration leukoencephalopathy malformation
NN-46	**hemiparesis** mastectomy pin somnolence bug dissociation exchange gain hemidiaphragmatic	NN-30	**hemiparesis** hemodilution hypoventilation infarction iodine laceration meclizine nimodipine ophthalmologist

Resulting subcategories created for each word from the previous approach sometimes was noisy. Therefore we decided to add various thresholds on the

occurrences of subcategories of each word. Thresholds used were 10, 50, 100, 200 based on the assumption that a big threshold would keep the subcategories for each word that appear many times in the corpus and will be more precise and informative. For each threshold three lexicons were created with the most frequent subcategories and the resulting attributes were added to the initial system.

3.6 Design of experiments

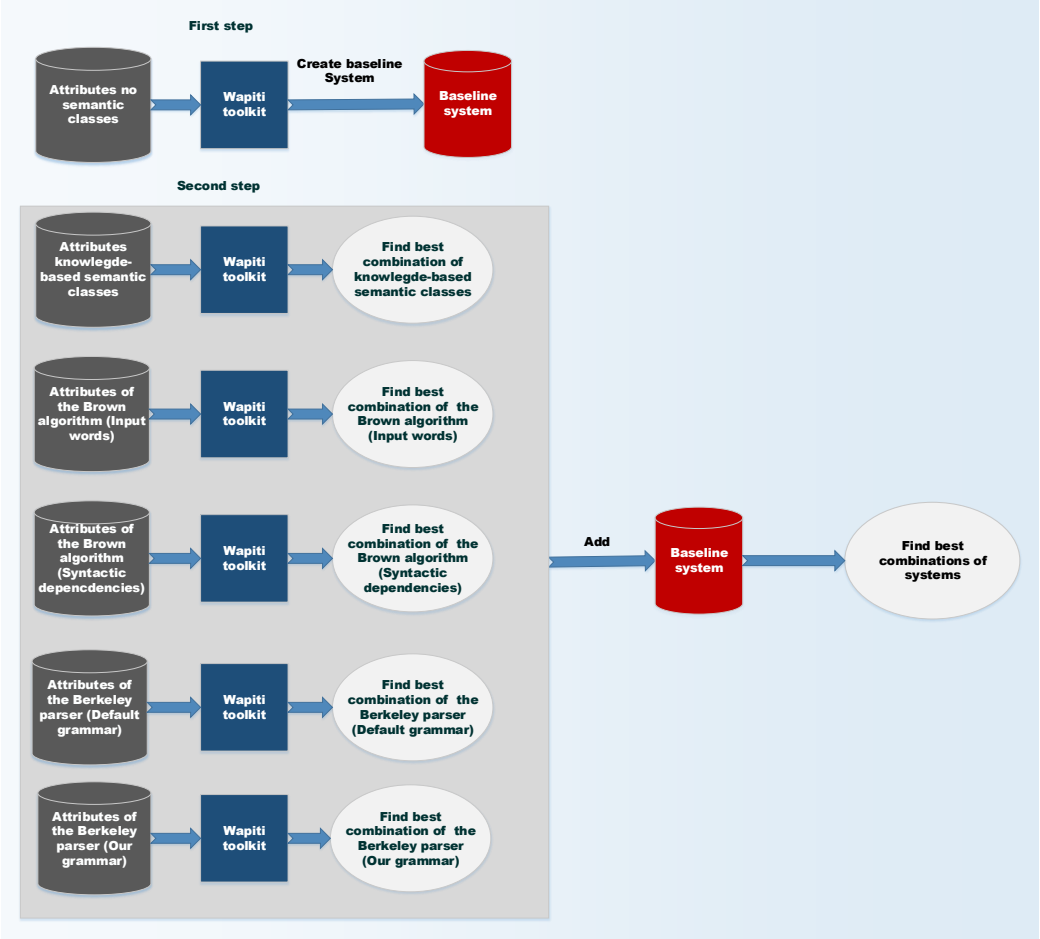

Figure 3.6 – Design of experiments general steps.

Figure 3.6 provides an overview of the design of our experiments which is based on three sequential steps. First step of the experiments was to find the best combination of attributes without knowledge-based semantic classes or data-driven word classes. Experiments were organized based on these two

type of classes. An automated script was used to create template files in order to add faster attributes combinations in Wapiti toolkit. Template given to Wapiti toolkit for each system described next in this chapter included unigrams, bigrams and both for each token, for one or two position before and after each focusing token.

UMLS attributes and Wikipedia attributes were added separately to the tabular form of the baseline system in most of the possible combinations to achieve the best F-measure performance. After the best combination of each was achieved both, UMLS and Wikipedia features, were added to of the baseline system to find the best combination.

Next step was to add attributes based on the Brown clustering algorithm with input string of words in order to compare them with our results. Attributes created were different combinations of minimum occurrences of the words and number of clusters. Each combination was rated separately on the baseline system and in the end combination of all the attributes was performed. Finally the Brown algorithm attributes were added in the tabular form that occurred from the addition of knowledge-based semantic classes to test the combinations of both types of attributes.

After achieving the best performance with the Brown clustering algorithm with input string of words, our approach with the syntactic dependencies were tested. For each input representation different combinations of minimum occurrences of the words and number of clusters were created. Resulting classes of the Brown clustering algorithm of the first approach of input representations described in Section 3.5.1 (page37) were added in the baseline system. Attributes were tested each one separately and then with various combinations with each other. Next step was to test the second approach described in Section 3.5.1 (page37). Resulting Brown classes for each of the three conjoined relation-token representations were added in the baseline system and tested separately and with each other. Each of the attributes resulting from input representation and the Brown clustering algorithm described above, were added in the system occurred from addition of knowledge-based semantic classes.

Finally, syntactic dependencies resulting from the Berkeley parser were tested. First step was to add in tabular form of the baseline system the output latent categories created by the parser with the provided default English grammar. Three attributes were created based on the top three latent categories of each word which have been rated separately and in combinations with each other. Next step was to rate the attributes created by our grammar, obtained by the Charniak-McClosky parser and the Berkeley parser. Three more attributes, based on the top three latent categories of each word, were created and added in the baseline system. Final step was to create attributes from the three top latent categories of each word of our grammar after applying the thresholds described in previous section. Three attributes for each threshold were created and added in tabular form of the baseline system.

Each of the above attributes were rated separately and with each other and also added in the resulting system after the addition of knowledge-based semantic classes and in the system with the Brown algorithm attributes.

Chapter 4

Evaluation

Contents

In Chapter 3 we described methods and algorithms used to improve name entity recognition with the aid of unsupervised learning of word classes. We focus on the processing steps of the algorithms, the required input data, the challenges encountered, as well as the design of the experiments used. In this chapter we provide the measurements and observations from our best experiments of these methods. This chapter is organized based on Chapter 3, results from knowledge-based semantics classes will be firstly presented, then results of data-driven word classes and finally the sum of the results. Final section of the Chapter is discussion about possible reasons of obtaining these results.

4.1 Results

For ease of use, results are structured in tables. All tables contain in first column the type of feature set, in second, third and forth column respectively "Precision", "Recall" and "F-measure" values. Each row represents a feature set and its best results based on the F-measure value. In each table of results, "no semantic classes" feature set refers to the best achieved result of baseline system described in Chapter 3. Best results of precision, recall and F-measure are marked with blue.

4.1.1 Knowledge-Based semantic classes

Table 4.1 – Disorder noun phrases evaluation of knowledge-based semantic classes: precision, recall and F-measure

Disorder noun phrases evaluation

Feature set	Precision	Recall	F-measure
Wikipedia features	79.03	24.32	37.20
UMLS features	78.22	63.31	69.98
No semantic classes	85.31	65.10	73.85
UMLS + Wikipedia features	78.98	64.50	71.01
No semantic classes + UMLS	88.10	74.23	80.57
No semantic classes + Wikipedia	86.58	68.02	76.18
No semantic classes + UMLS + Wikipedia	88.28	75.05	81.13

As we can see in Table 4.1 (page50) in first three lines, system with no semantic classes has the best F-measure, recall and precision when baseline system and semantic classes are used separately. The best achieved F-measure of knowledge-based semantic classes, which is 81.13 F-measure, 75.05 recall and 88.28 precision, is the result of a combined system of both semantic classes with the baseline system. Figure 4.1 provides a graph with the results described in Table 4.1 for a better understanding.

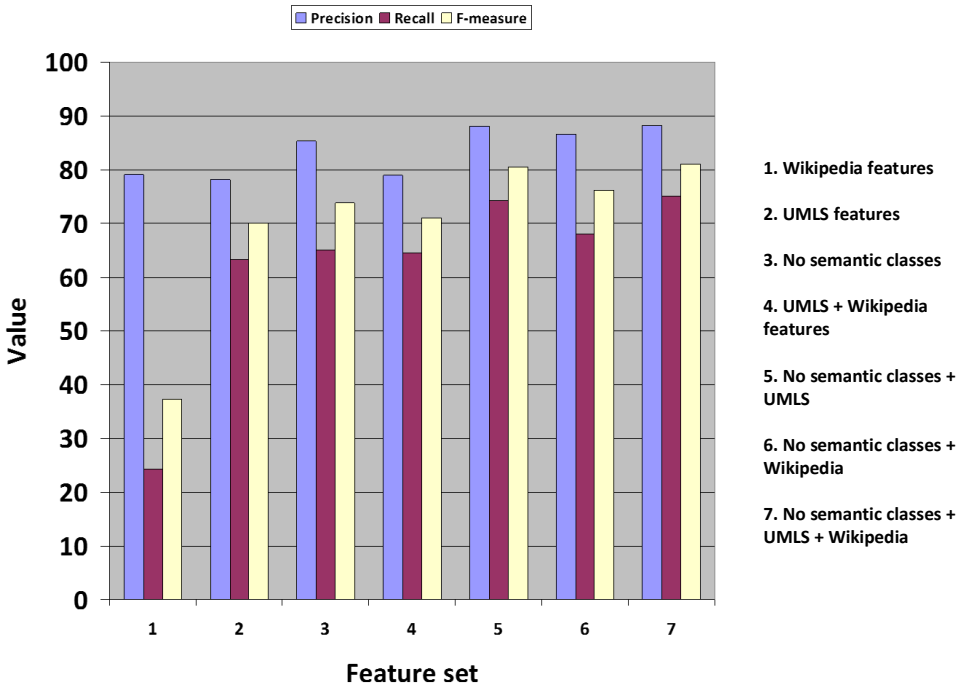

Figure 4.1 – Knowledge-based semantic classes results.

4.1.2 Data-driven word classes

Table 4.2 – Disorder noun phrases evaluation of data driven word classes : precision, recall and F-measure

Disorder noun phrases evaluation

Feature set	Precision	Recall	F-measure
Berkeley parser (default grammar)	72.86	59.45	65.47
Berkeley parser (our grammar)	74.61	62.57	68.06
No semantic classes	85.31	65.10	73.85
Brown clustering algorithm (string of words)	77.28	71.53	74.30
Brown clustering algorithm (syntactic dependencies)	78.65	71.74	75.04
No semantic classes + Berkeley parser (default grammar)	85.32	67.09	75.11
No semantic classes + Berkeley parser (our grammar)	85.53	67.72	75.59
Berkeley parser (default grammar) + Brown clustering algorithm (syntactic dependencies)	80.64	71.63	75.86
Berkeley parser (default grammar) + Brown clustering algorithm (string of words)	80.02	72.45	76.05
Berkeley parser (our grammar) + Brown clustering algorithm (string of words)	80.48	72.44	76.25
Berkeley parser (our grammar) + Brown clustering algorithm (syntactic dependencies)	81.21	73.20	77.00
No semantic classes + Brown clustering algorithm (syntactic dependencies)	85.52	71.72	78.01
No semantic classes + Brown clustering algorithm (string of words)	84.06	72.78	78.01
No semantic classes + Brown clustering algorithm (syntactic dependencies) + Berkeley parser (our grammar)	85.09	72.17	78.10
No semantic classes + Brown clustering algorithm (string of words) + Berkeley parser (Default grammar)	84.49	72.85	78.24
No semantic classes + Brown clustering algorithm (syntactic dependencies) + Berkeley parser (default grammar)	84.98	72.62	78.31
No semantic classes + Brown clustering algorithm (string of words) + Berkeley parser (our grammar)	85.33	73.47	78.96

Table 4.2 provides results of combinations of data-driven word classes. The first part of the table, above the horizontal line, shows that the Brown algorithm with syntactic dependencies achieves the highest precision, recall and F-measure when each type of attributes is tested separately. Second part of the table shows the combinations tested among data driven word classes. Best F-measure and recall were achieved by the combination of the baseline system with the Brown algorithm using as input string of words and the Berkeley parser using our grammar.

The Brown clustering algorithm using string of words and our approach when combined with the baseline system achieve the same F-measure, with syntactic dependencies approach to have a small difference of better precision but worse recall. Systems combined with attributes resulting from the Berkeley parser using our grammar achieved better F-measure than systems with attributes resulting from the parser with default grammar, except the system using no semantic classes, the Brown algorithm with syntactic dependencies and the Berkeley parser using the default grammar which achieved higher F-measure . In any case, differences are very small. Figure 4.2 provides a graph with the results described in Table 4.2 for a better understanding.

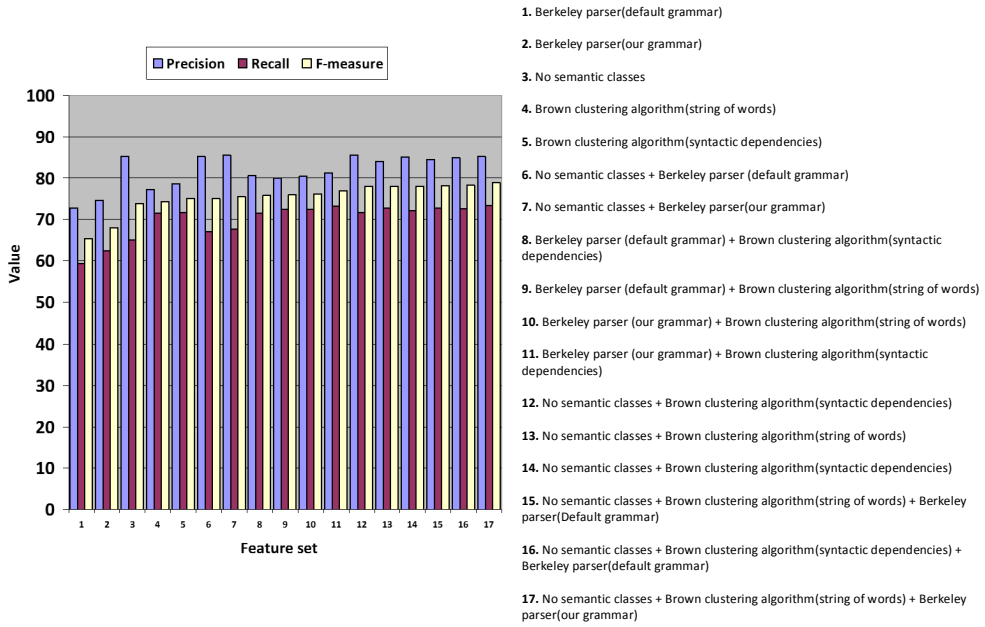

Figure 4.2 – Data-driven word classes results.

4.1.3 Combined results

Table 4.3 – Disorder noun phrases evaluation: precision, recall and F-measure

Disorder noun phrases evaluation

Feature set	Precision	Recall	F-measure
Wikipedia + Berkeley (default grammar)	77.35	63.26	69.60
UMLS + Berkeley (default grammar)	83.42	72.96	77.84
UMLS + Berkeley (our grammar)	83.89	73.37	78.28
No semantics + Wikipedia + Berkeley (our grammar)	86.41	71.72	78.38
UMLS + Wikipedia + Berkeley (default grammar)	84.16	73.61	78.54
UMLS + Wikipedia + Berkeley (our grammar)	84.70	74.01	78.99
UMLS + Brown clustering algorithm (string of words)	81.78	76.63	79.12
No semantics + Wikipedia + Brown clustering algorithm (string of words)	85.11	74.08	79.21
UMLS + Wikipedia + Brown clustering algorithm (string of words)	81.93	76.74	79.25
No semantics + Wikipedia + Brown clustering algorithm (string of words) + Berkeley (our grammar)	85.04	74.40	79.36
UMLS + Wikipedia + Berkeley (our grammar) + Brown clustering algorithm (string of words)	84.29	77.20	80.59
UMLS + Wikipedia + Berkeley (default grammar) + Brown clustering algorithm(string of words)	84.27	77.31	80.64
No semantics + UMLS + Berkeley (our grammar)	87.94	74.87	80.88
No semantics + UMLS + Wikipedia + Berkeley (our grammar)	88.13	75.69	81.44
No semantics + UMLS + Brown clustering algorithm (syntactic dependencies)	87.91	76.46	81.79
No semantics + UMLS + Brown clustering algorithm (string of words) + Berkeley(our grammar)	87.02	77.31	81.88
No semantics + UMLS + Wikipedia + Brown clustering algorithm (string of words) + Berkeley (default grammar)	87.33	77.15	81.92
No semantics + UMLS + Brown clustering algorithm (string of words)	87.22	77.28	81.95
No semantics + UMLS + Wikipedia + Brown clustering algorithm (syntactic dependencies)	88.09	76.63	81.99
No semantics + UMLS + Wikipedia + Brown clustering algorithm (string of words)	87.49	77.48	82.18
No semantics + UMLS + Wikipedia + Brown clustering algorithm (syntactic dependencies) + Berkeley (our grammar)	88.59	76.71	82.22
No semantics + UMLS + Wikipedia + Brown clustering algorithm (string of words) + Berkeley (our grammar)	87.55	77.80	82.39

Table 4.3 provides the results of combination of knowledge-based semantic classes and data-driven word classes. Results are provided in ascending order based on F-measure. Best combination was achieved by no semantic classes, knowledge based semantic classes, the Brown algorithm using as input string of words and the Berkeley parser using our grammar. Difference of F-measure with second ranking system is ~0.17, meaning that the differences between correctly categorized tokens is around six words. Second ranking system achieved higher precision but lower recall. The difference among them is that the second ranking system contains the attributes created by the Brown algorithm with syntactic dependencies instead of string of words. Next two systems in ranking which differ in attributes created by the Brown clustering algorithm also have a very small difference of F-measure. Attributes resulting from the Berkeley parser and our grammar in many cases add information to the system but differences in results are minor.

Second ranking system has lower F-measure, in comparison with first ranked system, because Wapiti toolkit based on the attributes of the Brown algorithm using syntactic dependencies categorized more tokens as positive cases of disorders which belonged to an outside entity as shown in Table 4.4:

Table 4.4 – Comparison of of the Brown algorithms classification result

Brown algorithm with syntactic dependencies			**Brown algorithm with input string of words**		
Token	**True class**	**Predicted class**	**Token**	**True class**	**Predicted class**
diagnostic	O	B-DISORDER	diagnostic	O	O
repolarization	O	I-DISORDER	repolarization	O	O
abnormalities	O	I-DISORDER	abnormalities	O	O

In order to test the differences between the top three ranked systems we computed a statistical significance using the Welch t-test. Between first and second system we obtained a p-value < 2.2e-16, which is statistically significant because p-value < 0,05. For first and third system we obtained also p-value < 2.2e-16 and for the second system with the third system we also obtained p-value < 2.2e-16. This indicates that differences between the systems are significant.

Figure 4.3 provides a graph with the results of UMLS attributes combined with attributes created by the Berkeley parser with the default grammar and with our grammar, and the top eight ranked systems provided in Table 4.3 (page 54) which underlines how small are the differences.

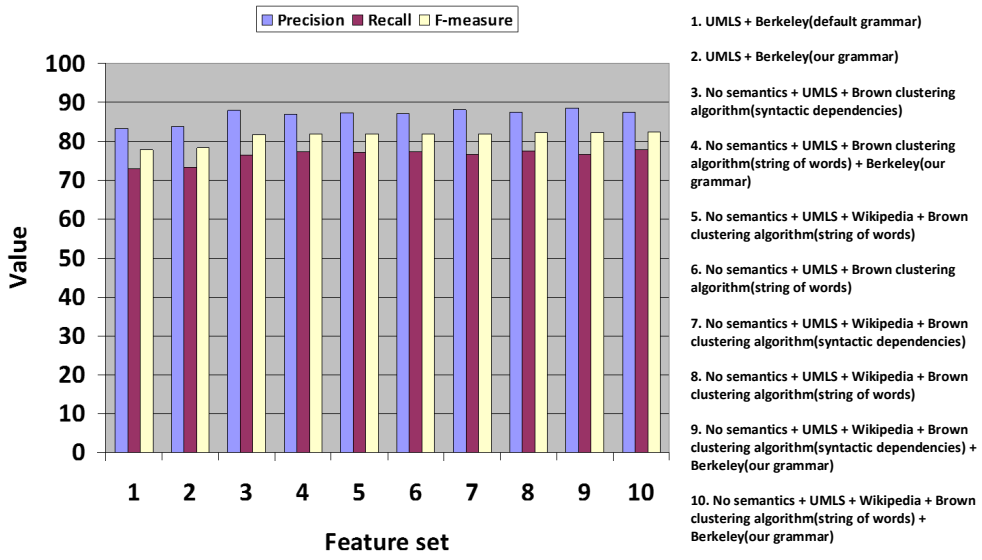

Figure 4.3 – Combined results.

Which semantic classes contribute most to the best result?

Given the best combination (No semantics, UMLS, Wikipedia, Brown clustering algorithm (string of words), Berkeley (our grammar)) which achieved precision 87.55, recall 77.80 and F-measure 82.39, if we remove the UMLS attributes we have a loss of ~3 points of F-measure and specifically precision 85.04, recall 74.40 and F-measure 79.36. If Wikipedia attributes are removed we only have a loss of ~0.5 points of F-measure. This indicates that UMLS attributes contribute more than Wikipedia attributes.

Does the retrained grammar improve the results?

Using the Berkeley parser with the default grammar instead of the Berkeley parser with our grammar we had a ~0.5 point of loss of F-measure by achieving precision 87.33, recall 77.15 and F-measure 81.92. Therefore retraining the Berkeley parser improved the final result by 0.5 points of F-measure. Also adding the latent categories obtained with the default grammar deteriorated the results because without it are higher by 0.3 points of F-measure (precision 87.49, recall 77.48 and F-measure 82.18).

What is the influence of the coverage of the word classes?

An important factor in understanding the systems is to measure coverage of the attributes on the corpus. Coverage will be examined as the proportion of tokens in the corpus for which a non-null value is provided by a lexicon for

a given set of word classes. Both knowledge-based semantic classes and data-driven semantic classes have a big amount of attributes. UMLS attributes are consisted of ten different attributes and Wikipedia from two attributes. Twenty attributes were created based on the Brown clustering algorithm with input string of words and twenty four based on the Brown algorithm with syntactic dependencies. Attributes created by the Berkeley parser are three for the default grammar and sixteen attributes for our grammar. From the above attributes not all were used in systems. But to compare the coverage we examined the coverage of each attribute. Table 4.5 provides coverage of the corpus per feature set and in each column contains a coverage of one attribute of each feature set. Because of the number of the created attributes we present only three values in the table. In first column the lowest coverage of the attributes is presented, in second column the median value of coverage of the attributes and in third column the highest coverage obtained by the feature set.

Attributes created by the Brown algorithm using as input string of words have the highest coverage ranging from 99.30%, 109506 tokens with assigned value, to 97.910%, 107931 tokens with assigned value. Attributes created by our grammar and the Berkeley parser have a coverage ranging from 94.10%, 103709 tokens with assigned value, to 63.16% ,69643 tokens with assigned value, mainly because of the use of thresholds. UMLS and Wikipedia attributes have the lower coverage of all attributes. Higher coverage is obtained by the Brown clustering algorithm using as input string of words which achieves to cover the corpus for 90 percentage and more.

In the sixth row the Berkeley parser using our grammar is described. Three features used in our best combinations is the top first and second category of latent categories with threshold 50 and the top first category of latent categories with threshold 100. In first column is coverage of the attribute resulting from the second top latent category of threshold 50, in second column is attribute resulting with threshold 100 and last column the attribute resulting from the first top latent category with threshold 50. Third column has a higher coverage than the second because of the threshold in occurrences of words, as higher the threshold as lower the number of words that are included in the lexicon.

Table 4.5 – Coverage of the corpus per feature set

	Attributes coverage of the corpus		
Features set	**Minimum**	**Median**	**Maximum**
UMLS	2.90%	16.40%	21.59%
Wikipedia	3.17%		4.80% [1]
Brown clustering algorithm(Syntactic dependencies)	74.50%	96.59 %	97.65%
Brown clustering algorithm(Input string of words)	97.91%	98.87%	99.30%
Berkeley parser(Default grammar)	83.60%	91.90%	95.53%
Berkeley parser(Our grammar)	63.16%	86.40%	94.08%

4.2 Discussion

Knowledge-based semantic classes have a small coverage on the corpus compared to data-driven word classes. As they are available for a small proportion they may not be often useful. On the other hand, data-driven word classes provide a higher coverage meaning that they will be often useful. However, they contain more noise than the knowledge-based classes. In this section we will evaluate our approach to test if data-driven semantic classes can improve named entity recognition and we will provide a comparison with knowledge-based word classes.

Are data-driven word classes as useful as knowledge-based word classes?

First we will evaluate if data-driven word classes are as useful as knowledge-based word classes. Brown algorithm attributes, when combined with each other achieve better F-measure than any combination of knowledge-based attributes. But in combination with the baseline system, UMLS and Wikipedia achieve precision 88.28, recall 75.05 and F-measure 81.13 compared to best combination of data-driven word classes which achieved precision 85.33, recall 73.47 and F-measure 78.96. Although data-driven classes have higher coverage than knowledge-based semantic classes they have lower results. A reason is that data-driven semantic classes are created automatically and errors occur and add noise to the system. Knowledge-based semantic classes are time-consuming but they achieved higher F-measure by two points. Given the above results data-driven word classes had achieved results with small differences compared to knowledge-based semantic classes but still did not achieve their F-measure.

Berkeley parser latent categories seen as refined POS categories

Another interesting point is that Berkeley parser latent categories are obtained by splitting syntactic categories into more specific subcategories. For this reason an extra experiment was created to compare part-of-speech(POS) attribute with the attributes created by the Berkeley parser. Berkeley parser using our grammar achieved precision 74.61, recall 62.57 and F-measure 68.06 where part-of-speech attribute achieved precision 49.58, recall 22.16 and F-measure 30.63. Difference is high indicating that Berkeley parser refined categories are much more informative than the standard part-of-speech categories.

1. There are only two Wikipedia attributes

Are data-driven word classes complementary to knowledge-based word classes?

Next important point to evaluate is if data driven word classes are complementary to knowledge-based classes. For the evaluation we will use as baseline the best achieved results of UMLS and Wikipedia attributes, precision 8.98, recall 64.50, F-measure 71.01. Data-driven classes when combined with knowledge-based semantic classes increase performance, as we can see in the first four rows of the Table 4.6. Addition of the no semantic attributes in the two classes created the highest results. The comparison of this results, provided in the last two rows of Table 4.6 indicates that one point of F-measure can be achieved by the addition of data-driven word classes.

Table 4.6 – Comparison of knowledge-based semantic classes with data driven word classes

Disorder noun phrases evaluation

Feature set	Precision	Recall	F-measure
UMLS + Berkeley (our grammar)	83.89	73.37	78.28
UMLS + Wikipedia + Berkeley (our grammar)	84.70	74.01	78.99
UMLS + Brown clustering algorithm (string of words)	81.78	76.63	79.12
UMLS + Wikipedia + Brown clustering algorithm (string of words)	81.93	76.74	79.25
no semantics + UMLS + Wikipedia	88.28	75.05	81.13
no semantic + UMLS + Wikipedia + Brown clustering algorithm (string of words) +Berkeley (our grammar)	87.55	77.80	82.39

Top three ranked systems in Table 4.3 (page 54) were obtained by a combination of both types and no semantics attributes indicating that they are complementary even if the increase of performance is not as high as we expected. Basic reason is that as performance increases is more difficult to obtain results with significant differences.

Which type of data-driven word classes is best to use?

Among data-driven word classes from combined results provided in Table 4.2 (page 52) and Table 4.3 (page 54) we can see that attributes created from Brown algorithm using syntactic dependencies when combined independently have higher performance than Brown algorithm on input words, but when combined with other attributes, Brown algorithm on input words achieve better results. Among attributes created by the Berkeley parser, attributes created by our grammar achieve higher performance. The Brown algorithm processing syntactic dependencies and the Berkeley parser using the default grammar and our grammar, require syntactic parsing over a large corpus which takes a long time. For this reason they have higher complexity

than the Brown algorithm applied on flat corpus. Among data-driven methods, the Brown algorithm applied on flat corpus achieved better results ,when combined with no semantic classes and knowledge based classes, and is the most simple considering the complexity.

Which type of knowledge-based semantic classes is best to use?

Finally among knowledge-based word classes UMLS attributes add more information to the system than Wikipedia attributes. Wikipedia attributes can be complementary to UMLS attributes but with only minimal improvement due to the fact that they have the least coverage among all attributes.

Chapter 5

Conclusions and Future Work

In this research we tested the use of unsupervised word classes in entity recognition and especially in named entity recognition. Data-driven word classes were added in a supervised machine learning system as additional information and were evaluated based on their performance and complexity.

5.1 Summary

Conditional random fields implemented by the Wapiti toolkit, annotated and unannotated data were used, among with two machine learning algorithms. The Brown clustering algorithm and the Berkeley parser were the two basic algorithms used to test the hypothesis that syntactic dependencies may add important information to the system and increase its performance.

We presented experiments based on different type of attributes. Initially we used the baseline system without semantic classes. We presented experiments based on knowledge-based semantic classes for the comparison of experiments with data-driven word classes. The Brown clustering algorithm processing string of words was one of the initial experiments on data-driven word classes since it is an efficient method which achieves high performance results.

Contribution of syntactic dependencies on named entity recognition were tested on two basic steps. Initially, the Brown clustering algorithm was used with the variation that instead of string of words, syntactic dependencies were used as input. Syntactic dependencies were created with Charniak-McClosky parser and were added in the Brown clustering algorithm. Different representations of the syntactic dependencies as input were tested and with different variables of the Brown algorithm.

Next step to add syntactic dependencies as additional information of entity recognition on disorders was based on the Berkeley parser. The Berkeley parser created annotated trees with syntactic categories based on a given grammar. Our methodology used the specific subcategories of the Berkeley

parser called latent categories as additional attributes in our named entity recognition system. Initially, we used the English default grammar provided by the parser but we also wanted to create these latent categories based on the biomedical domain. So we created a grammar which adopts the specializations of the biomedical domain with the Berkeley parser and used different threshold in order to remove the noisy subcategories.

We compared all of the above systems and also created combinations in order to obtain the higher possible performance. We presented the results of each system and each combination of systems combined with the coverage obtained by the created attributes. Knowledge-based semantic classes proved to be very important and achieved high results even though they lack of coverage and especially in a specialized domain like biomedical domain. Data-driven word classes achieved results with small differences compared to knowledge based semantic classes but still performed worse than knowledge-based semantic classes. Data-driven word classes had the higher coverage of the corpus but due to the unsupervised extracted information they added noise to the system. Knowledge-based semantic classes and data-driven word classes proved to be complementary by increasing the performance by one point. The Brown clustering algorithm with input string of words which has the lower complexity among the rest described data-driven word classes proved to obtain higher results.

5.2 Perspectives

In the future unsupervised word classes used on named entity recognition must be furtherly investigated. We would also like to test if the combination of Brown algorithm processing as input string of words with the Brown algorithm processing as input syntactic dependencies can add more information. Basic distributional methods to create word classes such as Grefenstette [27] should be tested. The Berkeley parser looks promising given our results especially as part-of-speech tagger. An interesting approach would be to test different thresholds to obtain the latent categories and ways to remove the noise created by the automated creation of classes. The advantage of the Berkeley parser is that latent categories are based on the grammar used by the algorithm. More research could be done regarding to the grammar and how it could add more accurate information adapted to the biomedical domain. Data-driven word classes are very promising and in future research an effort to reduce noise should be made in order to reach the same level as knowledge-based classes.

nadw poio apo ta duo einai swsto

Bibliography

[1] P. Liang, "Semi-supervised learning for natural language," Master's thesis, Massachusetts Institute of Technology, 2005. – Cited pages V, 13, 14, 24 et 38.

[2] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proc of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ser. ACL-44. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 433–440. [Online]. Available: http://dx.doi.org/10.3115/1220175.1220230 – Cited pages V, 1, 4, 16 et 17.

[3] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Computational Linguistics*, vol. 19, pp. 313–330, 1993. – Cited pages V, 26 et 27.

[4] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, Dec. 1992. [Online]. Available: http://dl.acm.org/citation.cfm?id=176313.176316 – Cited pages V, 4, 15, 16 et 72.

[5] B. Tang, H. Cao, Y. Wu, M. Jiang, and H. Xu, "Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features." *BMC Med Inform Decis Mak*, vol. 13 Suppl 1, p. S1, 2013. [Online]. Available: http://www.biomedsearch.com/nih/Recognizing-clinical-entities-in-hospital/23566040.html – Cited pages 1, 9 et 10.

[6] C. Sutton and A. Mccallum, "Introduction to conditional random fields for relational learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2006. – Cited pages 4, 8 et 22.

[7] H. Suominen, S. Salanter'a, W. W. Sumitra Velupillai Chapman, G. Savova, N. Elhadad, S. Pradhan, B. R. South, D. L. Mowery, G. J.

Jones, J. Leveling, L. Kelly, L. Goeuriot, D. Martinez, and G. Zuccon, "Overview of the ShARe/CLEF eHealth evaluation lab 2013," in *Proceedings of CLEF 2013*, ser. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2013, to appear. – Cited pages 4 et 20.

[8] A. Bodnari, L. Deléger, T. Lavergne, A. Névéol, and P. Zweigenbaum, "A supervised named-entity extraction system for medical text," in *Proc of CLEF 2013*, 2013, to appear. – Cited pages 4, 16, 32 et 33.

[9] S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training," in *Proc of 2004 Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, vol. 4, 2004, pp. 337–342. – Cited pages 4, 12 et 13.

[10] B. de Bruijn, C. Cherry, S. Kiritchenko, J. D. Martin, and X. Zhu, "Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010." *JAMIA*, vol. 18, no. 5, pp. 557–562, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/jamia/jamia18.html#BruijnCKMZ11 – Cited pages 4 et 12.

[11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. [Online]. Available: http://dl.acm.org/citation.cfm?id=645530.655813 – Cited page 8.

[12] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale crfs," in *Proc of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 504–513. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858733 – Cited pages 8 et 23.

[13] T. G. Dietterich, "Machine learning for sequential data: A review," in *Proc of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. London, UK, UK: Springer-Verlag, 2002, pp. 15–30. [Online]. Available: http://dl.acm.org/citation.cfm?id=645890.671269 – Cited page 8.

[14] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel, "Improvements that don't add up: ad-hoc retrieval results since 1998," in *Proc of the 18th ACM conference on Information and knowledge management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 601–610. [Online]. Available: http://doi.acm.org/10.1145/1645953.1646031 – Cited page 8.

[15] "Conll: the conference of signll." Website, http://ifarm.nl/signll/conll/. – Cited page 8.

[16] O. Uzuner, I. Solti, and E. Cadag, "Extracting medication information from clinical text," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 514–518, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1136/jamia.2010.003947 – Cited pages 9 et 20.

[17] J. Patrick and M. Li, "A cascade approach to extracting medication events," in *Proc of the Australasian Language Technology Association Workshop 2009*, Sydney, Australia, Dec. 2009, pp. 99–103. [Online]. Available: http://www.aclweb.org/anthology/U09-1014 – Cited page 9.

[18] R. Leaman and G. Gonzalez, "Banner: An executable survey of advances in biomedical named entity recognition." in *Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, T. Murray, and T. E. Klein, Eds. World Scientific, 2008, pp. 652–663. [Online]. Available: http://dblp.uni-trier.de/db/conf/psb/psb2008.html#LeamanG08 – Cited page 9.

[19] S. Petrov and D. Klein, "Improved inference for unlexicalized parsing," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proc of the Main Conference*. Rochester, New York: Association for Computational Linguistics, April 2007, pp. 404–411. [Online]. Available: http://www.aclweb.org/anthology/N/N07/N07-1051 – Cited page 10.

[20] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proc of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 384–394. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858721 – Cited page 12.

[21] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proc of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 160–167. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390177 – Cited page 12.

[22] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model." in *NIPS*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1081–1088. – Cited page 12.

[23] A. Sun, R. Grishman, and S. Sekine, "Semi-supervised relation extraction with large-scale word clustering," in *Proc of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies - Volume 1, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 521–529. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002539 – Cited page 12.

[24] X. Zhu, C. Cherry, S. Kiritchenko, J. Martin, and B. De Bruijn, "Detecting concept relations in clinical text: Insights from a state-of-the-art model," *J. of Biomedical Informatics*, vol. 46, no. 2, pp. 275–285, Apr. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jbi.2012.11.006 – Cited page 12.

[25] M. Johnson and A. E. Ural, "Reranking the berkeley and brown parsers," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 665–668. [Online]. Available: http://dl.acm.org/citation.cfm?id=1857999.1858094 – Cited page 14.

[26] M. Collins and T. Koo, "Discriminative reranking for natural language parsing," *Comput. Linguist.*, vol. 31, no. 1, pp. 25–70, Mar. 2005. [Online]. Available: http://dx.doi.org/10.1162/0891201053630273 – Cited page 14.

[27] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *Proc of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, ser. HLT-NAACL '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 152–159. [Online]. Available: http://dx.doi.org/10.3115/1220835.1220855 – Cited page 14.

[28] G. Grefenstette, "Corpus-derived first, second and third-order word affinities," in *Proc of EURALEX*, 1994. – Cited pages 15 et 62.

[29] S. O. Petrov, "Coarse-to-fine natural language processing," Ph.D. dissertation, EECS Department, University of California, Berkeley, Aug 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-116.html – Cited page 16.

[30] "University of pittsburgh nlp repository," Website, http://www.dbmi.pitt.edu/nlpfront. – Cited page 20.

[31] O. Uzuner, I. Goldstein, Y. Luo, and I. Kohane, "Identifying patient smoking status from medical discharge records." *Journal of the American Medical Informatics Association : JAMIA*, vol. 15, no. 1, pp. 14–24, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1197/jamia.m2408 – Cited page 20.

[32] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark, "Multiparameter intelligent monitoring in intensive care ii: a public-access intensive care unit database." *Critical care medicine*, vol. 39, no. 5, pp. 952–960, May 2011. [Online]. Available: http://dx.doi.org/10.1097/ccm.0b013e31820a92c6 – Cited page 20.

[33] L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," in *Proc of the Third Annual Workshop on Very Large Corpora*. ACL, 1995, pp. 82–94. – Cited page 22.

[34] B. Gu, "Recognizing nested named entities in genia corpus," in *Proc of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, ser. BioNLP '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 112–113. [Online]. Available: http://dl.acm.org/citation.cfm?id=1567619.1567642 – Cited page 22.

[35] B. Alex, B. Haddow, and C. Grover, "Recognising nested named entities in biomedical text," in *Proc of BioNLP*, 2007, pp. 65–72. – Cited page 22.

[36] Q. T. Zeng, S. Goryachev, S. Weiss, M. Sordo, S. N. Murphy, and R. Lazarus, "Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system," *BMC Medical Informatics and Decision Making*, vol. 6, no. 1, pp. 30–39, Jul. 2006. [Online]. Available: http://dx.doi.org/10.1186/1472-6947-6-30 – Cited page 22.

[37] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE-07-001, 2007. – Cited page 30.

[38] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications." *Journal of the American Medical Informatics Association : JAMIA*, vol. 17, no. 5, pp. 507–513, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1136/jamia.2009.001560 – Cited page 32.

[39] "Umls:unified medical language system," Website, http://www.nlm.nih.gov/research/umls/. [Online]. Available: http://www.nlm.nih.gov/research/umls/ – Cited page 33.

[40] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program." National Library of Medicine, National Institutes of Health, Bethesda, MD

20894, USA. alan@nlm.nih.gov, 2001, pp. 17–21. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/11825149 – Cited page 34.

[41] D. McClosky and E. Charniak, "Self-training for biomedical parsing," in *Proc of the Association for Computational Linguistics (ACL 2008, short papers)*. Columbus, Ohio: The Association for Computer Linguistics, 2008, pp. 101–104. [Online]. Available: http://dblp.uni-trier.de/db/conf/acl/acl2008s.html#McCloskyC08 – Cited pages 37 et 43.

[42] M. catherine De Marneffe and C. D. Manning, "Stanford typed dependencies manual," 2008. [Online]. Available: http://nlp.stanford.edu/software/dependencies_manual.pdf – Cited page 37.

[43] T. Koo, X. Carreras, and M. Collins, "Simple semi-supervised dependency parsing," in *Proc of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 595–603. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1068 – Not cited.

[44] M. F. Mahbub Chowdhury and A. Lavelli, "Disease mention recognition with specific features," in *Proc of the 2010 Workshop on Biomedical Natural Language Processing*, ser. BioNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 83–90. [Online]. Available: http://dl.acm.org/citation.cfm?id=1869961.1869972 – Not cited.

[45] H. M. Wallach, "Conditional random fields: An introduction," University of Pennsylvania, Tech. Rep., 2004. – Not cited.

[46] H. Zhang, M. Zhang, C. L. Tan, and H. Li, "K-best combination of syntactic parsers," in *Proc of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, ser. EMNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 1552–1560. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699648.1699702 – Not cited.

Chapter 6

Organisation

6.1 LIMSI

The Computer Sciences Laboratory for Mechanics and Engineering Sciences (LIMSI) is a CNRS laboratory associated with UPMC and Paris-Sud Universities, based in Orsay France. It involves multidisciplinary research in Mechanical and Chemical Engineering and in Sciences and Technologies for Information and Communication. The research fields have a wide range from cognition to acoustics,spoken language and text processing etc.

6.2 ILES

The Information, Language, written and Signed (ILES) group research interests focus in language modeling, natural language processing (mostly morphological and semantic), including sign language processing.

Chapter 7

Related work

Table 7.1 – Phrase-level horizontal evaluation: overall, narrative, and list

		Phrase-level horizontal evaluation								
		Overall			**Narrative**			**List**		
Rank	Group	P	R	F	P	R	F	P	R	F
1	USyd	0.896	0.820	0.857	0.685	0.63	0.656	0.914	0.835	0.873
2	Vanderbit	0.840	0.803	0.821	0.571	0.606	0.588	0.901	0.814	0.855
3	Manchester	0.864	0.766	0.812	0.692	0.542	0.608	0.858	0.805	0.831
9	UofUtah	0.832	0.715	0.769	0.504	0.531	0.517	0.859	0.657	0.744
10	UWinsconsinM	0.904	0.661	0.764	0.366	0.405	0.384	0.931	0.51	0.659

Table 7.2 – Performance for feature accumulations in the Relations Task

Feature set	**Recall**	**Precision**	**F-score**
(a) Baseline	0.646	0.718	0.680
(b) +order/type-sensitive	0.672	0.731	0.700
(c) +rich word features	0.681	0.753	0.715
(d) +domain knowledge	0.694	0.750	0.721
(e) +syntax	0.694	0.763	0.727
(f) +unannotated data	0.693	0.773	0.731

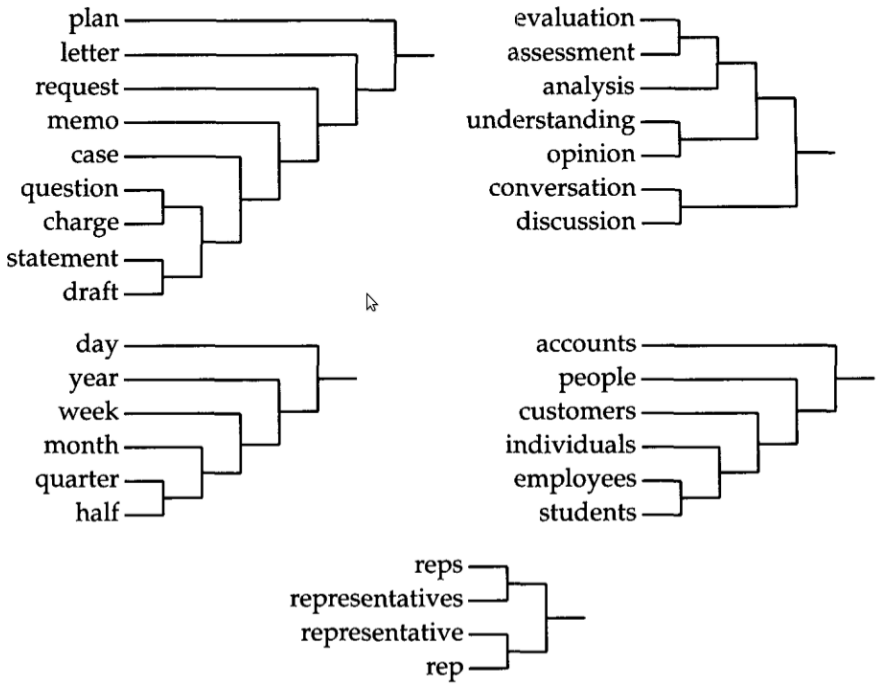

Figure 7.1 – Sample subtrees from a 1,000-word mutual information tree [4]
.

Figure 7.1 shows an example of subtrees of Brown clustering.

Chapter 8

Data-driven word classes

Input representation	Class	Tokens included in class
original text files	11111111110011111	nurse ileostomy wh PEG face pancreatitis pacemaker SBP GI drain **stroke**
	111111101111011	disc main anterior **hemiparesis**
token relation token	11111101111111	endocarditis dementia rectum radiation trauma sepsis seizure **stroke** bleeding fluid mass
	111111001001	aphasia tachypnea hemorrhoids toxicity ketoacidosis hyponatremia hypothyroidism nephropathy obesity **hemiparesis** flare thrombocytopenia
relation-token token	011110001	syndrome ischemia hernia stroke dysfunction bruits lymphadenopathy **stroke** erythema Diabetes
	1111100100	prep_from_shifts **conj_vs_stroke** prep_of_staging conj_but_anemia prep_due_to_liver conj_or_hypercholesterolemia
	1111100100	**prep_due_to_stroke** conj_and_hypothryoidism prep_for_Respiratory prep_from_distension conj_and_Glidewire prep_with_myocardium

	1111100100	appos_thinning conj_hydronephrosis prep_as_alkalosis **prep_per_stroke** prep_with_fibroids xsubj_hypertension dobj_02-09 prep_with_Pseudoaneurysm
	011110010	**hemiparesis** run contacts dysarthria hypoglycemia tachypnea dysphagia tightness lethargy
	1110000111	**num_hemiparesis** num_lymphocyte num_implant num_Dysphagia num_valvuloplasty dep_0x18mm
	1110111110	partmod_Caroline partmod_atrium **partmod_hemiparesis** dep_HYPERVENTILATION conj_and_suicidal partmod_flagyl
	1110111111	**rcmod_hemiparesis** dep_significance conj_but_persisted conj_but_surgery dep_diseases xcomp_unchanged dep_CARDIOLOGY
	111100111	**amod_hemiparesis** nsubj_bruits amod_cysts amod_erythema amod_defects amod_contusion
	1111011011	amod_dysrythmias **nsubj_hemiparesis** prep_of_relaxation prep_on_forming amod_Ligation conj_and_atheroma partmod_healthy prep_for_Coumadinized
token token - relation	011101001101	inflammation necrosis renal involvement pathology acute compression trauma **stroke** ischemia abscess infarct hemorrhage cephaloid
	101111011100	root_palpable root_DVT root_distention root_crackles root_ventricle root_intervention **root_stroke** root_membranes root_ischemia root_region
	11011111101	pobj_questions pobj_metoprolol pobj_ischemia pobj_workup pobj_return **pobj_stroke** pobj_attempt pobj_pre pobj_w pobj_fact pobj_BID

	111010010	dobj_wound dobj_sensation dobj_abdomen dobj_lobe dobj_areas dobj_effect dobj_laceration dobj_examination dobj_presence dobj_consolidation **dobj_stroke** dobj_valve dobj_infarct dobj_UTI
	1110100111	tmod_tobacco prep_for_drinks advcl_smokes num_SI dobj_POA **prep_prior_to_stroke** dobj_NASH prep_that_fevers partmod_unaided dobj_nightly
	1110101000	prep_into_ureter prep_from_hepatic prep_of_parameters prep_from_arteries prep_compared_to_prior **prep_in_stroke**
	111010100111	dobj_07-03 prep_with_spontaneous **agent_stroke** conj_and_resultant prep_after_recent prep_with_atypia prep_with_plastic prep_with_intracranial prep_as_factor prep_with_thoracic
	011101001100	neglect paralysis claudication aphasia flexion emphysema **hemiparesis** neuropathy
	101111011100	root_dermatitis root_aphasia root_Evaluation root_infiltration root_septal **root_hemiparesis** root_rupture
	111010010	nsubj_Hypothyroidism nsubj_retention nsubj_Hyponatremia nsubj_extubation **nsubj_hemiparesis** nsubj_radiology
	1111111000	appos_dilatation **conj_and_hemiparesis** prep_of_lumen prep_of_SVC prep_to_colon
token token	11101100	disorder Anemia sepsis seizure PE ischemia **stroke** Disease rash aspiration ucler hypotension GI anemia

110111010	cholecystitis appendage complex deformity perforation hydrocephalus **hemiparesis** infarcts emphysema necrosis views dilation

Table 8.1 – Examples of Brown clusters obtained based on representations of syntactic dependencies.

Index