Building a Named Entity Recognizer in Three Days: Application to Disease Name Recognition in Bulgarian Epicrises

Georgi D. Georgiev
Ontotext AD
IT Center Office Express, 3rd floor, 135
Tsarigradsko Shosse, Sofia 1784, Bulgaria
georgi.georgiev@ontotext.com

Valentin Zhikov
Ontotext AD
IT Center Office Express, 3rd floor, 135
Tsarigradsko Shosse, Sofia 1784, Bulgaria
valentin.zhikov@ontotext.com

Borislav Popov
Ontotext AD
IT Center Office Express, 3rd floor, 135
Tsarigradsko Shosse, Sofia 1784, Bulgaria
borislav.popov@ontotext.com

Preslav Nakov
National University of Singapore
Department of Computer Science
13 Computing Drive, Singapore 117417
nakov@comp.nus.edu.sg

Abstract

We describe experiments with building a recognizer for disease names in Bulgarian clinical epicrises, where both the language and the domain are different from those in mainstream research, which has focused on PubMed articles in English. We show that using a general framework such as GATE and an appropriate pragmatic methodology can yield significant speed up of the manual annotation: we achieve F1=0.81 in just three days. This is the first step towards our ultimate goal: named entity normalization with respect to ICD-10.

1 Introduction

The problems of named entity recognition and normalization are central to biomedical text processing: as part of the typical preprocessing pipeline, they are key for any deep text analysis.

The goal is to identify all mentions of named entities of a particular type, e.g., genes, proteins, diseases, drugs, and to propose a canonical name, or a unique identifier, for each mention. Solving this problem is important for many applications, e.g., enriching databases such as the *Protein and Interaction Knowledge Base*[1] (PIKB), part of *LinkedLifeData*, compiling gene-disease-drug search indexes for large document collections in *KIM*[2] (e.g., using the BioMedicalTagger[3]) and *MEDIE*[4], or building a search engine that can retrieve the effects of a drug on various diseases in research papers and patents.

Being so central to biomedical text processing, the problems of named entity recognition and normalization have received a lot of research attention, e.g., there have been several related competitions at BioNLP[5] and BioCreAtIvE[6]. Moreover, high-quality manually annotated biomedical text corpora such as GENIA[7] have been created, which have enabled the development of a number of biomedical text processing tools that need such kind of data for training.

Unfortunately, mainstream research has so far focused almost exclusively on English and on biomedical abstracts and full-text articles in PubMed. Thus, biomedical named entity recognition (NER) for languages other than English or for other types of biomedical texts faces the problem of the lack of manual text annotations and biomedical resources in general, which are needed for machine learning. While manually annotating some data is always a good idea, e.g., for analysis, parameter tuning and evaluation, it is hardly practical for more than just a few documents. It is thus important to make smart use of any existing resources and to facilitate the process of manual annotation as much as possible so that good results can be achieved quickly and with very little efforts. The best approach depends on the particular task as well as on the kinds of texts and resources that are available, and there is hardly a universal solution. Still, there are probably lessons to be learned from particular examples of efforts focusing on achieving good performance for NER in new languages and domains in a short period of time.

[1] http://www.linkedlifedata.com
[2] http://www.ontotext.com/kim
[3] http://www.ontotext.com/life-sciences/semantic-biomedical-tagger
[4] http://www-tsujii.is.s.u-tokyo.ac.jp/medie/
[5] http://sites.google.com/site/bionlpst/
[6] http://biocreative.sourceforge.net/
[7] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi

Proceedings of the Workshop on Biomedical Natural Language Processing, pages 27–34,
Hissar, Bulgaria, 15 September 2011.

Below we describe our experience with building a recognizer for disease names in Bulgarian clinical epicrisis, where both the language and the domain are different from those in mainstream research. We demonstrate good performance with little efforts and in a very short period of time. We further show how we can save about 57% of the efforts needed for manual annotation of instances of named entities in text. Finally, we discuss how this work can be extended to the task of named entity normalization.

The remainder of the paper is organized as follows: Section 2 offers an overview of related work, Section 3 summarizes our methodology, Section 4 describes our experiments and presents the results, Section 5 goes into deeper analysis, Section 6 describes some potential applications, and Section 7 concludes with possible directions for future work.

2 Related Work

There have been several research efforts focused on making manual annotations over a text corpus or building a system for named entity recognition and information extraction in a short period of time and with limited resources.

Ganchev *et al.* (2007) proposed a semi-automated approach to named entity annotation where first an *n*-best MIRA-based named entity recognizer is trained on the initial training set and then tuned for high recall by manipulating the MIRA loss function. Then, its output is checked by a human annotator, who makes yes/no decisions for each proposed entity. Their experiments show that this can speed up manual annotation by about 58% without loss in quality of annotation. We achieve a similar reduction of 57% in the required annotation time using a structured perceptron for named entity tagging; however, we start with no annotated data at all.

Settles (2011) described the DUALIST system for semi-supervised annotation based on active learning. The system solicits and learns from labels on both features (e.g., words) and instances (e.g., entities). It has been evaluated on a number of annotation and classification tasks; on named entity recognition, it achieved 0.80 precision (unknown recall and F1), which is a bit lower than our 0.86-0.87 precision. Moreover, being based on active learning, DUALIST needs access to a large number of unlabeled documents from which to choose examples for annotation; such documents are not available in our case.

Freedman *et al.* (2011) presented a bootstrapping system for what they call *extreme extraction*, where they start with an ontology defining the target concepts and relations they will need to extract and a limited number of training data. They achieve human-level accuracy in a week; this includes five hours of manual rule writing. In fact, our case is arguably more extreme since we start with no annotated data at all.

We should also mention the early work done as part of the Surprise Language Exercises held in 2003, where sixteen teams tried to develop language technologies for two previously unanticipated languages, Cebuano and Hindi, in just ten and twenty-nine days, respectively. This work is described in two special issues of the *ACM Transactions on Asian Language Processing* journal (Oard, 2003).

Finally, we should mention the 2007 Computational Medicine Challenge[8], which focused on analyzing clinical epicrises but for English. However, it asked for assigning ICD-9 codes at the document level, while we want to find *instances* of *IDC-10* disease mentions in text. Moreover, the challenge provided a lot of manually curated data, and thus there was no need to annotate additional data (Crammer et al., 2007).

3 Method

We started with a small number of Bulgarian epicrises and a list of diseases from an ontology.

First, we analyzed and manually annotated a small number of documents to acquaint ourself with the data and the task and to produce datasets for development and evaluation.

Next, we automatically induced contextual rules for finding additional names of diseases. We applied these rules on the development set, we inspected their output, and we incrementally restricted them in several iterations. Once we were satisfied with the precision, we applied the rules to new texts, and we collected their predictions to build a gazetteer of likely disease names.

Next, we applied the gazetteer to some unannotated documents, and we inspected and corrected the matches in context, thus ending up with more annotated documents for training. We then trained a sequence-based named entity recognizer on all training documents we had so far.

Finally, we augmented that sequence recognizer with the predictions of the gazetteer as features, thus achieving 0.81 F1 score, which we found sufficient, and we stopped there.

[8] http://computationalmedicine.org/challenge/previous

4 Experiments and Evaluation

4.1 Initial Datasets

We started with a collection of 100 Bulgarian documents describing clinical epicrises, which we analyzed manually. Based on this analysis, we developed annotation guidelines, which we used to annotate 20 of these documents with the names of diseases from the ICD-10[9] (International Classification of Diseases) ontology. There were a total of 441 disease names mentioned in these 20 documents.

4.2 Contextual Rule Induction

We selected 10 of the annotated documents for development, and used the remaining 10 documents for testing.

We automatically learned contextual rules from the development set, which we then used to find named entities in the test set.

The rules memorize three tokens to the left and to the right around a potential disease name instance. Here is an example (`???` marks the target instance):

```
diagnosis : ??? . Polyneuropathia Diabetica
```

We do not allow the context words to cross sentence boundaries, and thus the inferred rules can have access to a context of less than three words on either or both sides of `???`, as in the following example:

```
the therapy of ??? .
```

Here are some inferred rules extracted from the original text in Bulgarian:

```
ДИАГНОЗА : ??? . ПОЛИНЕВРОПАТИЯ ДИАБЕТИКА

. ??? . ХИПОТИРЕОИДИЗМУС АУТОИМУНЕС

. ??? . ХИПЕРТОНИЯ АРТЕРИАЛИС

. II . ??? . ПИЕЛОНЕФРИТИС ХРОНИКА

. ??? .

във   връзка   със   ???   ,   диагностично
уточняване

степента на ??? и лечение .

Минали заболявания : ??? от 20 години

стабилна стенокардия , ??? от 25 години

млада възраст , ??? и еритема нодозум

5 години , ??? – хипотиреоидна фаза

ЕМГ данни за ??? . Намалена скорост

за лечение на ??? . Проведе се

в терапията на ??? .
```

Rules with a balanced context of three words on either side proved to be restrictive and very reliable. We indexed the annotated documents in GATE (Cunningham, 2000) so that we could perform fast search for annotations and context words on either side of a disease mention.

The rules were implemented in JAPE format[10]:

```
Rule: One
Priority: 100(
     ({Token.string   == "ДИАГНОЗА"})
     ({Token.string   == ":"})
     (({Token})+):bind
     ({Token.string   == "."})
-->
     :bind.PreDisease = {rule = "One"}
```

The preceding rule states that if the word "diagnosis" is followed by ":", all following tokens up to and not including "." should be considered part of a disease name.

Figure 1 shows the user interface for searching and visualization of the results in the context of three tokens to the left/right of the candidate disease names. Fast searching allows us to find that two tokens on the right hand side and only one on the left hand side could yield better results.

4.3 Inducing a Disease Gazetteer

We executed the rules on the 80 unannotated documents and we created a gazetteer based on the recognized disease names. The resulting gazetteer was evaluated based on its ability to find correct disease mentions in text, on the development set and on the test set. The results are shown in Table 1.

	R	**P**	**F1**
Dev set	0.61	0.30	0.40
Test set	0.61	0.49	0.54

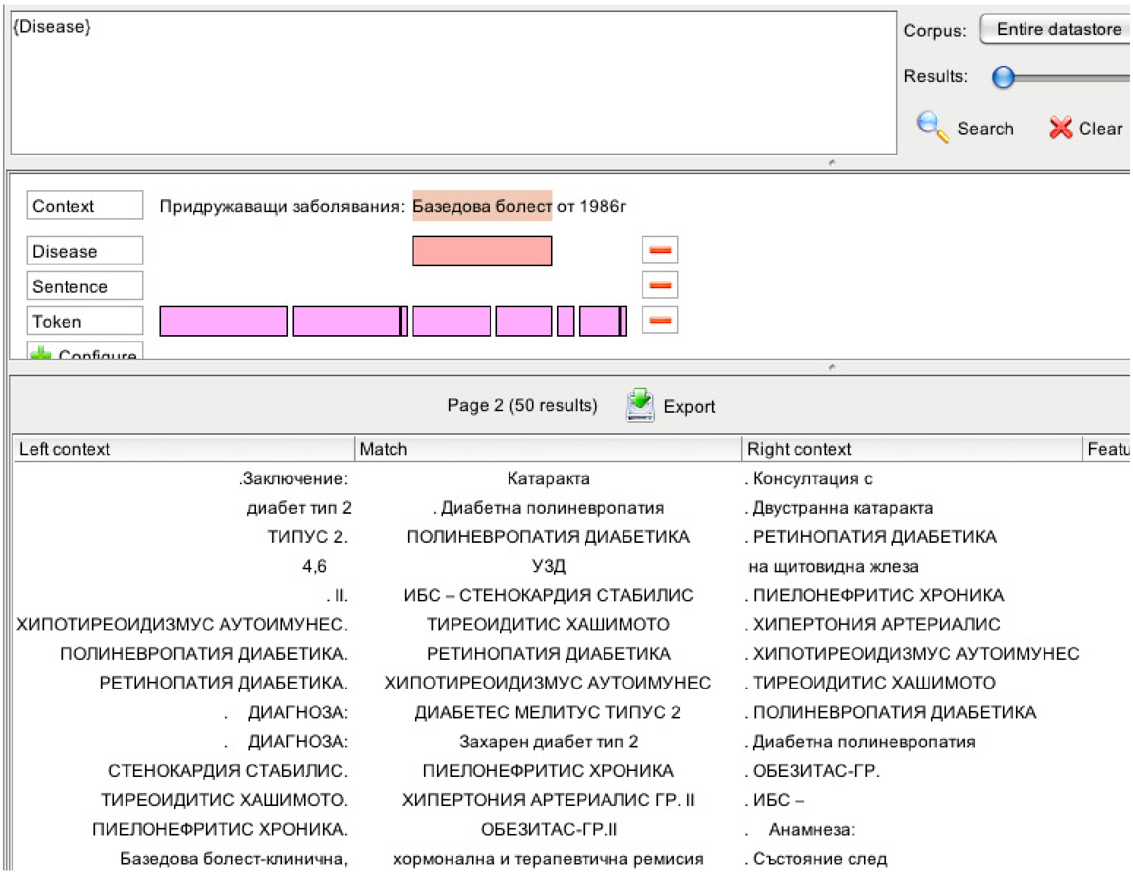
Table 1: Evaluation of the gazetteer that was induced using the context rules.

As Table 1 shows, the low precision is a more important problem for the induced gazetteer than low recall. We thus focused on improving precision by adding rules that could filter out some bad extracted candidates.

We first experimented with length-based filters, removing all candidates whose length is less than *n* symbols; we tried *n* = 5, 6, 7, 8, 9, 10. The results are shown in Table 2. We can see that using length filters significantly increases precision without negatively affecting recall: precision jumps from 0.3 to 0.65, which is higher than recall. As a result, F1 raises from 0.4 to 0.61.

[9] http://www.who.int/classifications/icd/en/

[10] http://gate.ac.uk/sale/tao/index.html#x1-2030008



{Disease}

Придружаващи заболявания: Базедова болест от 1986г

Page 2 (50 results)

Left context	Match	Right context	Featu
.Заключение:	Катаракта	. Консултация с	
диабет тип 2	. Диабетна полиневропатия	. Двустранна катаракта	
ТИПУС 2.	ПОЛИНЕВРОПАТИЯ ДИАБЕТИКА	. РЕТИНОПАТИЯ ДИАБЕТИКА	
4,6	УЗД	на щитовидна жлеза	
. II.	ИБС – СТЕНОКАРДИЯ СТАБИЛИС	. ПИЕЛОНЕФРИТИС ХРОНИКА	
ХИПОТИРЕОИДИЗМУС АУТОИМУНЕС.	ТИРЕОИДИТИС ХАШИМОТО	. ХИПЕРТОНИЯ АРТЕРИАЛИС	
ПОЛИНЕВРОПАТИЯ ДИАБЕТИКА.	РЕТИНОПАТИЯ ДИАБЕТИКА	. ХИПОТИРЕОИДИЗМУС АУТОИМУНЕС	
РЕТИНОПАТИЯ ДИАБЕТИКА.	ХИПОТИРЕОИДИЗМУС АУТОИМУНЕС	. ТИРЕОИДИТИС ХАШИМОТО	
. ДИАГНОЗА:	ДИАБЕТЕС МЕЛИТУС ТИПУС 2	. ПОЛИНЕВРОПАТИЯ ДИАБЕТИКА	
. ДИАГНОЗА:	Захарен диабет тип 2	. Диабетна полиневропатия	
СТЕНОКАРДИЯ СТАБИЛИС.	ПИЕЛОНЕФРИТИС ХРОНИКА	. ОБЕЗИТАС-ГР.	
ТИРЕОИДИТИС ХАШИМОТО.	ХИПЕРТОНИЯ АРТЕРИАЛИС ГР. II	. ИБС –	
ПИЕЛОНЕФРИТИС ХРОНИКА.	ОБЕЗИТАС-ГР.II	. Анамнеза:	
Базедова болест-клинична,	хормонална и терапевтична ремисия	. Състояние след	

Figure 1: The user interface used to manually annotate instances.

Filter	R	P	F1
Length < 5	0.61	0.47	0.53
Length < 6	0.61	0.54	0.57
Length < 7	0.60	0.60	0.60
Length < [8-10]	0.58	0.65	0.61
+ "^[0-9] .*"	0.56	0.72	0.62
+ specific words	0.56	0.87	0.68

Table 2: Evaluation (on the *development dataset*) of the gazetteer that was induced from context rules *and additional filtering rules*.

Next, we looked closer at the development dataset, and we found that many candidate disease names that started with numbers were in fact false positives. Thus, we tried adding a regular expression like "^[0-9] .*" as an additional filter to the length filter of 8. This improved precision from 0.58 to 0.72, but recall dropped from 0.60 to 0.56, and thus, F1 increased only slightly: from 0.61 to 0.62.

We further tried removing candidates containing certain words like *menopause* and *unencumbered*, which our manual analysis has found to give rise to many false positives. Doing so led to another increase in precision to 0.87 on the development set while keeping the recall intact at 0.56. The F1 score for the gazetteer after these rules were applied increased from 0.62 to 0.68 (R=0.56, P=0.87) on the development set. The same F1 of 0.68 was also achieved on the test set, but there recall and precision were more balanced: R=0.64, P=0.73.

These results are arguably not strong enough for a system that recognizes disease mentions in clinical epicrisis in a fully automated fashion. Still, the resulting gazetteer could potentially be useful for a number of tasks, e.g., for making more manual annotations faster. It could also help robustness since it captures most of the rules that were defined in the annotation guidelines created during the first run of the manual annotation: the 20 documents that we used for development and testing.

Thus, we decided to use the gazetteer to help the annotation of 10 more documents. The annotation required about 1.6 minutes per document for the one annotator and 3.6 minutes for another one, or 2.6 minutes on average. As a comparison, in the first annotation run, the average time for annotating a document was about 6 minutes for both annotators. This is a 57% reduction in the time needed to annotate a new document.

There was also an improvement in the quality of the manual annotation process. The average agreement between the gazetteer and the manual annotators was F1=0.84 (R=0.88, P=0.80) for the first annotator and 0.77 (R=0.67, P=0.91) for the second one. The inter-annotator agreement was F1=0.78 (R=0.88, P=0.70). This is comparable to the inter-annotator agreement between the two annotator on the first 10 documents where F1 was 0.80 (R=0.96, P=0.69) and to the second 10 where F1 was 0.77 (R=0.91, P=0.67).

4.4 Training a Structured Perceptron for Disease Name Recognition in Text

The resulting 30 documents (20 for training and 10 for testing) were used to train a structured perceptron (Freund and Shapire, 1999). This learning algorithm was selected for simplicity and because of its fast online training. We used a standard set of features that has been initially proposed by McDonald & Pereira (1996), and then successfully adapted to Bulgarian by Georgiev *et al.* (2009); shown in Table 3.

Using this feature set, the perceptron achieved an F1 of 0.69, which is only slightly better than the 0.68 F1 of the rules/gazetteer approach.

In order to improve the performance, we annotated 10 more documents by first applying the gazetteer and then doing manual annotations to create System 1, which was trained on 30 documents and tested on 10. We further built System 2, which was trained as System 1, but also used matches with the gazetteer as features.

The results for System 1 in Table 4 show that adding more training data (i.e., 30 instead of 20 documents) yields only minor improvement in F1: from 0.69 to 0.71. However, also using features from the gazetteer in System 2 causes F1 to jump to 0.81. As we can see, this is due to a huge improvement in recall, which goes from 0.59 to 0.76, while precision remains stable.

We can conclude that the gazetteer turned out to be an important information source, probably because it had analyzed more text (90 documents; all but the testing 10 ones), and thus it could help recall a lot.

Predicate	**Regular Expression**
Initial capital	[А-Я].*
Capital, then any	[А-Я].
Initial capitals, alpha	[А-Я][а-я]*
All capitals	[А-Я]+
All lowercase	[а-я]+
Capitals mix	[А-Яа-я]+
Contains a digit	.*[0-9].*
Single digit	[0-9]
Double digit	[0-9][0-9]
Natural number	[0-9]+
Real number	[-0-9]+[\.,]?[0-9]+
Alpha-numeric	[А-Яа-я0-9]+
Roman	[ivxdlcm]+|[IVXDLCM]+
Contains dash	.*-.*
Initial dash	-.*
Ends with dash	.*-
Punctuation	[,\.;:\?!-+"]
Multidots	\.\.+
Ends with dot	.*\.
Acronym	[А-Я]+
Lonely initial	[А-Я]\.
Single character	[А-Яа-я]
Quote	["’]

Table 3: The orthographic predicates used by the structured perceptron named entity recognizer. The observation list for each token includes a predicate for each regular expression that matches it.

System	**R**	**P**	**F1**
System 1	0.59	0.87	0.71
System 2	0.76	0.86	0.81

Table 4: Evaluation of the structured perceptron. System 1 is trained on 30 documents and tested on 10. System 2 is trained like System 1, but it also uses features based on matches with the gazetteer.

5 Discussion

The experiments above have shown that manually annotating data and building a system for named entity recognition in clinical epicrises written in Bulgarian is hard for a number of reasons, including but not limited to the following:

(i) limited and chaotic general purpose text analysis resources for Bulgarian,
(ii) our lack of experience with such texts,
(iii) specificity of the domain language,
(iv) specificity of the terminology,
(v) specificity of the document structure,
(vi) issues with extracting the text from the Microsoft Word format the epicrises were stored in.

We should note that extracting disease names from epicrises would hardly have been much simpler for English, despite the existence of many biomedical corpora and tools for that language. The main problem here is the domain shift: the existing tools and resources for English are targeting almost exclusively journal papers, whose format, structure and vocabulary differs substantially from those of clinical epicrises.

On the positive side, we have shown that even though the task looks complicated, it could be solved with usable F1 in just three days.

This speed of building our system would have hardly been possible without our extensive use of the GATE framework for natural language engineering, which has saved us a lot of time and efforts. Among its features that have helped us the most were (*i*) its ability to extract text from Microsoft Word documents, (*ii*) its default Unicode tokenizer and (*iii*) its sentence splitter based on simple regular expressions, which we were able to adopt very quickly, thus overcoming the lack of general purpose text analysis tools and resources for our kind of biomedical text.

We were further able to speed up the process of manual annotation by focusing on rules based on words/tokens rather than on part of speech or lemmata (for which we did not have ready tools that could handle the domain well). This was possible because of the particular structure of the documents and the specific language use.

For example, clinicians tend to express the diagnosis at the beginning of the epicrisis, typically, in a paragraph that starts with the pattern "Diagnosis:" (or "Диагноза:" in Bulgarian). Here is an example:

ДИАГНОЗА: **Захарен диабет-тип 2. Артериална хипертония. Дислипидемия.**

The diagnosis is followed by few paragraphs explaining why and how the patient was examined, which is further followed by additional information about how the presence of the disease was tested.

Of course, a diseases can be extracted from other parts of the document, e.g., such that provide information about the examination of the patient by another specialist. For example, the structured paragraph below contains a list of diseases that have been suggested after a consultation with a neurologist:

Консултация с невролог: **Начален полиневропатен синдром.** Терапия: контрол на кръвната захар.

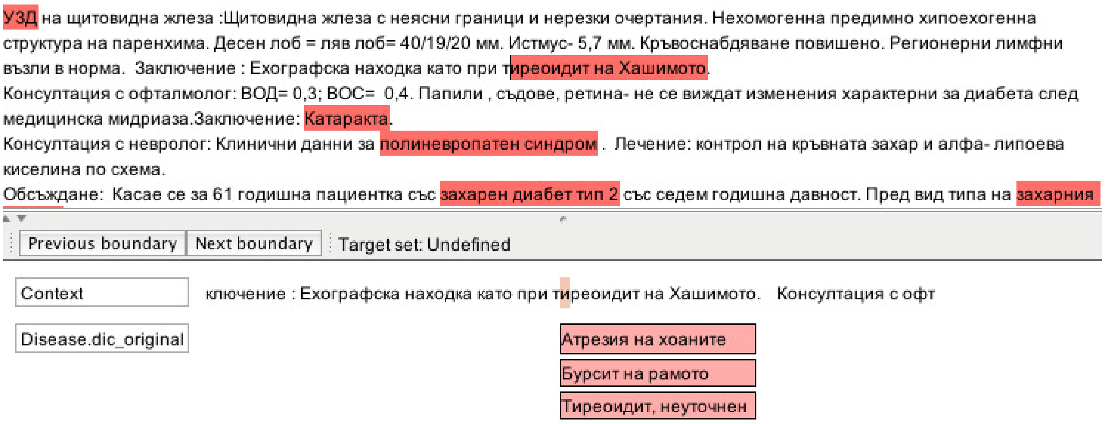
We should note that disease names can be mentioned not only in the diagnosis-related section(s) of an epicrisis, but can occur pretty much anywhere in the document. While catching all instances is generally hard, we were able to do it with the high F1 of 0.81 to a great extent because of the gazetteer. This is because most of the disease names mentioned outside of the diagnoses are likely to repeat those that have been already listed in the diagnosis section. Thus, once the gazetteer has been populated with the somewhat easy-to-extract disease names from the diagnosis-related sections, it can help find further instances of those disease names in contexts that are generally much more ambiguous. We have seen this effect above when comparing System 1 and System 2 in Table 4.

6 Potential Applications

As we have seen above, System 2 recognizes disease mentions in text with an F1 of 0.81, which is quite high and is arguably already usable for a number of practical applications. Still, generally, named entity recognition is just the first step in biomedical text analysis; we might also need normalization, which would allow us to get to the canonical names of the diseases mentioned in a particular epicrisis, thus enabling more sophisticated practical applications. For example, if a disease recognizer is coupled with a recognizer of dates and symptoms, we would be able to monitor disease progression and manifestation over time.

Normalizing disease names to an identifier or a canonical name in an ontology, would also allow linking a particular clinical epicrisis to a whole web of linked data. One such example would be *LinkedLifeData*, which is a platform that integrates biomedical information for diseases, symptoms, proteins, genes, drug action information and clinical trials. Linking between an epicrisis and LinkedLifeData might facilitate knowledge acquisition and enrichment and could enable sophisticated queries and rich semantic search over a collection of epicrises.

Thus, in order to enable such semantic annotations, we need not only the offsets and type of each disease mention in a given epicrisis but also a mapping of the mention to a unique identifier. An obvious candidate in our case is the Bulgarian version of the ICD-10 ontology, which provides both unique disease name identifiers and canonical forms that can be used for disease name normalization.



УЗД на щитовидна жлеза :Щитовидна жлеза с неясни граници и нерезки очертания. Нехомогенна предимно хипоехогенна структура на паренхима. Десен лоб = ляв лоб= 40/19/20 мм. Истмус- 5,7 мм. Кръвоснабдяване повишено. Регионерни лимфни възли в норма. Заключение : Ехографска находка като при тиреоидит на Хашимото.
Консултация с офталмолог: ВОД= 0,3; ВОС= 0,4. Папили , съдове, ретина- не се виждат изменения характерни за диабета след медицинска мидриаза.Заключение: Катаракта.
Консултация с невролог: Клинични данни за полиневропатен синдром . Лечение: контрол на кръвната захар и алфа- липоева киселина по схема.
Обсъждане: Касае се за 61 годишна пациентка със захарен диабет тип 2 със седем годишна давност. Пред вид типа на захарния

Target set: Undefined

Context: ключение : Ехографска находка като при тиреоидит на Хашимото.   Консултация с офт

Disease.dic_original:
Атрезия на хоаните
Бурсит на рамото
Тиреоидит, неуточнен

Figure 2: The GATE user interface for choosing the correct canonical names for the disease instances in text.

Motivated by the practical importance of the task, we did some preliminary assessment of the feasibility of the idea of mapping disease name mentions to identifiers in the ICD-10 ontology. Unfortunately, we found that this was not as simple as we thought initially since the disease names used in ICD-10 strongly disagreed with the names used in our clinical epicrises.

One reason for this is the tendency of Bulgarian clinicians to describe their diagnoses in Latin. Unfortunately, the Bulgarian ICD-10 does not include disease names in Latin. Moreover, there were many abbreviations, both for Latin and Bulgarian. Thus, for this task, it is important to collect Latin medical terminology from other sources as well as synonyms of diseases for Bulgarian, which can be used to enrich the ICD-10 disease classification. It is worth mentioning that we partially handle this problem by automatically generating a gazetteer of diseases from the source documents.

We further found that we needed to remove a number of identifier references from the ICD-10 names. In particular, we filtered out any disease names that looked like codes and were in parentheses, e.g., (J99.8*), (L40.5ї), (M45-M46ї, M48.-ї, M53-M54ї), (E10-E14ї с общ четвърти знак .4) and the like. We further filtered out some abbreviations that did not refer to the target disease but to molecular markers or abnormal proteins and other participants that can cause the disease. Here are some examples:

```
G36.0 Оптиконевромиелит [болест на Devic]

G36.1 Остър и подостър хеморагичен левкоенцефалит [болест на Hurst]
```

In order to increase the number of actual disease names for which we could provide ICD-10 identifiers and to create additional synonyms for some of the diseases, we reordered and selectively extracted some names in parentheses from the existing disease names in ICD-10 (rather than the description) of the disease in such cases. For example, we rewrote the two examples above as follows:

```
G36.0 Оптиконевромиелит

G36.0 болест на Devic

G36.1 Остър и подостър хеморагичен левкоенцефалит

G36.1 болест на Hurst
```

In order to automatically prepare the corpus for manual annotation of disease mentions, we created a GATE processing tool that implements a number of string distance metrics based on *SimMetric*, an open source library of similarity and distance metrics, including Levenshtein, L2, Cosine, Jaccard, Jaro-Winkler, etc. *SimMetric* has a visual interface, which facilitates the selection of the most appropriate similarity measure for a particular task. After some preliminary experiments, we found Jaro-Winkler to be most fit for our data.

Based on the score of the distance match between a disease mention in the text and the names in the ICD-10 dictionary, we ordered and select the top-3 names from ICD-10. We then fed these top-3 candidates in a GATE user interface, specially created for the purpose, which is shown on Figure 2.

In Figure 2, the text and the diseases are shown in the upper part of the screen, while the disease names from ICD-10 with the top-3 Jaro-Winkler scores are shown in the bottom. A human annotator can delete the candidates that are incorrect in the given context with a single mouse click, e.g., “УЗД”, which is a procedure/examination. In some rare cases, the annotator might also need to add new candidates, which can be done with a right click: see the case of “тиреоид на Хашимото”, where the correct candidate is “E06.3 Автоимунен тиреоидит” instead of “E06.9 Тиреоидит, неуточнен” that is present in the top 3 candidates.

7 Conclusion and Future Work

In this work we focus on simple approaches to named entity recognition having limited or no prior example data. In this framework, we have demonstrated that a seemingly complicated named entity recognition task can be handled with satisfying quality in a fast and robust manner. We have achieved this by examining the structure and language expressions, as well as words and orthographic features found in clinical epicrises. We have further demonstrated that using general purpose frameworks such as GATE and an appropriate pragmatic methodology can significantly speed up the process of annotation.

Our disease mentions recognizer, annotation guidelines and annotated epicrises are potentially useful for applications such as document categorization and search. Moreover, extending the disease mention recognition to semantic annotations with identifiers from an ontology such as ICD-10 would enable a number of applications such as monitoring disease progression and manifestation over a period of time and linking epicrises to a web of linked data like *LinkedLifeData*. We believe these are promising research directions and we plan to pursue them in future work.

For the purpose of facilitating and speeding up manual semantic annotations, we have developed a new GATE-based processing tool that can calculate string similarity scores between disease mentions found in text and disease names listed in ICD-10. We have further coupled this with a GATE visual resource that allows a human annotator to delete wrong mentions at a given offset, thus only leaving the correct option(s), while also allowing the addition of more options. In future work, we plan to use this interface in a similar pragmatic approach to the task of normalizing disease names in context with respect to ICD-10.

Acknowledgements

The research presented in this paper is partially supported by grant DO 02-292/December 2008 funded by the Bulgarian National Science Fund.

We would like to thank the anonymous reviewers for their detailed and constructive comments, which have helped us improve the paper.

References

Hamish Cunningham. 2000. *Software Architecture for Language Engineering*. PhD thesis, University of Sheffield, Sheffield, UK.

Marjorie Freedman, Lance Ramshaw, Elizabeth Boschee, Ryan Gabbard, Gary Kratkiewicz, Nicolas Ward and Ralph Weischedel. 2011. Extreme Extraction — Machine Reading in a Week. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'2011),* pp. 1437–1446, Edinburgh, Scotland, UK.

Yoav Freund and Robert Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277-296.

Kuzman Ganchev, Fernando Pereira, Mark Mandel, Steven Carroll and Peter White. Semi-Automated Named Entity Annotation. 2007. In *Proceedings of the Linguistic Annotation Workshop (LAW'07)*. pp. 53-56, Prague, Czech Republic.

Georgi Georgiev, Preslav Nakov, Kuzman Ganchev, Petya Osenova, Kiril Simov. 2009. Feature-Rich Named Entity Recognition for Bulgarian Using Conditional Random Fields. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP'2009)*. pp. 113-117, Borovets, Bulgaria.

Ryan McDonald and Fernando Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields, *BMC Bioinformatics*, 6(Suppl 1):S6 doi:10.1186/1471-2105-6-S1-S6

Douglas W. Oard. 2003. The Surprise Language Exercises. *ACM Transactions on Asian Language Processing*, 2(2):79–84.

Burr Settles. 2011. Closing the Loop: Fast, Interactive Semi-Supervised Annotation With Queries on Features and Instances. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'2011),* pp. 27–31, Edinburgh, Scotland, UK.

Koby Crammer, Mark Dredze, Kuzman Ganchev, Pratim Partha Talukdar and Steven Carroll. 2007. Automatic Code Assignment to Medical Text. *In Workshop on biological, translational, and clinical language processing (BioNLP)*, pp. 129-136, Prague, Czech Republic.