COMPARISON AND COMBINATION OF NAMED ENTITY RECOGNITION TOOLS APPLIED TO BIOGRAPHIC TEXTS

(İSİMLİ VARLIK TANIMA ARAÇLARININ BİYOGRAFİK MAKALELERE UYGULANARAK KARŞILAŞTIRILMASI VE BİRLEŞTİRİLMESİ)

by

Samet Atdağ, B.S.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

INSTITUTE OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisors: Dr. Vincent Labatut

June 2013

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided me the possibility to complete this thesis. A special gratitude I give to my thesis advisor, Vincent Labatut, whose contribution in useful comments, remarks, stimulating suggestions and encouragement, helped me to coordinate and finish my project. I appreciate his vast knowledge and skill in many areas, and his invaluable assistance in writing publications and this thesis.

Last but not least, I would like to thank my family, for giving birth to me at the first place and supporting me spiritually throughout my life. They were always supporting me and encouraging me with their best wishes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Named entity recognition (NER) is a popular domain of natural language processing. For this reason, many tools exist to perform this task. Amongst other points, they differ in the processing method they rely upon, the entity types they can detect, the nature of the text they can handle, and their input/output formats. This makes it difficult for a user to select an appropriate NER tool for a specific situation. In this work, we try to answer this question in the context of biographic texts. For this matter, we first correct, clean and complete the corpus constituted by B. Kupelioglu (KÜPELİOĞLU, 2012). We then select 4 publicly available, well known and free for research NER tools for comparison: Stanford NER, Illinois NET, OpenCalais NER WS and Alias-i LingPipe. We take advantage of the framework developed by Yasa Akbulut to compare Stanford, Illinois and OpenCalais, and complete it so that it can also handle LingPipe, too. We also add to this platform a new way of evaluating NER performance. We apply the NER tools to our corpus, assess their performances and compare them. When considering overall performances, a clear hierarchy emerges: Stanford has the best results, followed by LingPipe, Illionois and OpenCalais. However, a more detailed evaluation, performed relatively to entity types and article categories, highlights the fact their performances are diversely influenced by those factors. This complementarity brings us to the definition of a combination method, allowing to merge the results outputted by these individual tools in order to improve the overall performance. We realize this combination thanks to a Support Vector Machine (SVM) trained on our corpus. We also manually define a set of rules to perform the same operation, in order to have a baseline when assessing the performance of our combination tool. We have found that these rules are better at performing full detection of entities, but that the SVM classifier is better at performing partial detection of entities.

Keywords: Named Entity Recognition (NER), Tool Evaluation, Biographic Texts.

RESUME

La reconnaissance d'entités nommées (NER) est un domaine populaire du traitement du langage naturel. Pour cette raison, de nombreux outils existent pour réaliser cette tâche. Ils diffèrent entre autres par la méthode de traitement sur laquelle ils sont basés, les types d'entités qu'ils peuvent reconnaitre, la nature du texte qu'ils peuvent traiter, et les formats d'entrée/sortie. Pour une situation donnée, ceci rend la sélection d'un outil approprié difficile à effectuer pour l'utilisateur final. Dans ce travail, nous tentons de répondre à cette question dans le contexte de textes biographiques. Pour cela, nous corrigeons, nettoyons et complétons d'abord le corpus constitué par B. Kupelioglu (KÜPELİOĞLU, 2012). Nous sélectionnons ensuite 4 outils de NER accessibles librement, bien connus de la communauté, et gratuits quand utilisés pour la recherche académique. Nous profitons de la plateforme développée par Y. Akbulut, qui est capable de traiter Stanford NER, Illinois NET et OpenCalais WS. Nous la complétons pour qu'elle supporte aussi Alias-i LingPipe, ainsi que de nouvelles mesures de performance, adaptées à nos objectids. Nous applicons les 4 outils à notre corpus, puis évaluons et comparons leurs performances. En ce qui concerne les performances globales, une hiérarchie nette apparait : Stanford obtient les meilleurs résultats, suivi par LingPipe, Illionois et finalement OpenCalais. Cependant, une évaluation plus détaillée, réalisée relativement aux types des entités et aux catégories des textes, met en lumière le fait que les performances sont plus ou moins influencées par ces 2 facteurs. Cette complémentarité nous amène à définir une méthode de combinaison, permettant de fusionner les sorties des outils pris individuellement, afin d'améliorer la performance globale. Cette combinaison est réalisée grâce à une machine à vecteur de support (SVM), entrainée sur notre corpus. Nous définissons manuellement un ensemble de règles permettant de réaliser la même opération, afin d'avoir une référence pour évaluer la performance de notre outil de combinaison. Après évaluation, il

retourne que le système à base de règles est plus performant pour la détection complète d’entités, alors que le SVM est meilleur pour la détection partielle d’entités.

Mots-Clés: Reconnaissance d’entités nommées (NER), Outil d’évaluation, Textes biographiques.

ÖZET

İsimli varlık tanıma işlemi (İVT) doğal dil işleme alanında önemli bir bileşendir. Bu önemden ötürü, İVT işlemini gerçekleştiren çok sayıda araç bulunmaktadır. Bu araçlar, diğer farklarının yanısıra, dayandıkları işleme yöntemleri, tespit edebildikleri varlık tipleri, işleyebildikleri metinlerin yapısı ve girdi/çıktı formatları gibi özellikleri ile birbirlerinden ayrışmaktadırlar. Bu durum, kullanıcılar için İVT araçları arasında seçim uygun bir aracın seçilimini güçleştirmektedir. Bu çalışmada, biyografi metinleri kullanarak uygun bir İVT aracının nasıl seçilmesi gerektiğini incelemeyi hedefledik. Bunu başarabilmek için, öncelikle B. Küpelioğlu ((KÜPELİOĞLU, 2012)) tarafından oluşturulmuş olan metin kümesini düzenledik, temizledik ve eksik kalan kısımlarını tamamladık. Sonrasında kamuya açık, iyi bilinen ve bedava olan şu 4 İVT aracını seçtik: earch NER tools for comparison: Stanford NER, Illinois NET, OpenCalais NER WS and Alias-i LingPipe. Stanford, Illinois ve OpenCalais'i karşılaştırırken, Yasa Akbulut ((Akbulut, 2010)) tarafından geliştirilmiş olan altyapı kullanıldı ve bu çalışma dahilinde bu altyapıya LingPipe desteğini ekledik. İVT araçlarının performans değerlendirmesi için gerekli olan yeni bir değerlendirme yöntemini de ekleyerek altyapının bu konudaki eksiğini tamamladık. Ardından İVT araçlarını elimizdeki metin setine uygulayarak, performans değerlendirmelerini çıkardık ve karşılaştırdık. Sonuçlara baktığımızda, İVT araçları genel performans üzerinden iyiden kötüye doğru şu sıralamaya sahipler: Stanford, LingPipe, Illinois ve OpenCalais. Öte yandan, varlık tiplerini ve biyografilerin ait oldukları kategorileri de göz önüne alarak daha detaylı bir inceleme yaptığımızda araçların performansların bu etkenlere bağlı olarak farklılıklar arzettiklerini gözlemledik. Bu da bizi, belirli durumlarda daha iyi sonuçlar veren araçları birleştirerek daha iyi bir performans elde etme noktasına götürdü. Metin kümesi üzerinden eğittiğimiz bir destek vektör makinasını kullanarak bu birleştirme işlemini gerçekledik. Ardından aynı işlemi elle tanımladığımız belirli kurallar

üzerinden tekrarlayarak, otomatik birleştiricinin performansını test ettik. Birleştirme işlemi sonucunda oluşan yeni aracın performansına ait sonuçları gözlemledik.
Anahtar Sözcükler: İsimli Varlık Tanıma, Araç Performans Analizi, Biyografik Metinler.

1 INTRODUCTION

The work presented in this paper is part of a longer-term project, consisting in extracting a social network from events identified automatically from biographical articles available on Wikipedia. In this social network, nodes represent the individuals concerned and the links between them are obtained by integrating the events over a period of time. The extraction of the events itself is based on the identification of certain entities. We describe an event in several aspects, mainly: actors, objects, time and space. The project therefore consists of several steps: 1) identify the entities, 2) find the corresponding events and finally 3) build the social network. Our work is the last part of the first step, it focuses on Named Entity Recognition (NER). NER consists in detecting certain types of entities in a sentence, such as names of persons, places and organizations, Our work is the continuation of some works previously conducted by Y. Akbulut (Akbulut, 2010) and B. Kupelioglu (KÜPELİOĞLU, 2012).

Chronologically, the first of these works is that of Y. Akbulut (Akbulut, 2010), who set up a platform called Nerwip (*Named Entity Recognition on Wikipedia Pages*) allowing to apply several NER tools on texts extracted from Wikipedia. However, this platform suffers from several limitations. First, some problems exist regarding the application of the NER tools to a full corpus. Second, the supported NER tools is not representative enough of the existing tools. Third, the evaluation method leads to measures which are difficult to interpret. One of our tasks was to fix these problems and complete the platform.

The work by B. Kupelioglu (KÜPELİOĞLU, 2012) consisted in defining a tool able to recognize dates. More importantly, she also constituted a corpus of 249 biographical articles from Wikipedia, and processed them by hand, in order to annotated persons, locations, organizations and dates. The goal of this corpus was to conduct a large-scale evaluation of available NER tools, using Nerwip. However, there was some problems

with this corpus, such as the as text encoding and mislocated entities. Fixing the corpus in order to perform a reliable evaluation of the NER tools was another of our tasks.

In his work, Y. Akbulut also proposed to combine the outputs of several classifiers in order to get a better overall performance. However, in the absence of appropriate evaluation results, the proposed combination method was relying on a very raw, intuition-based manual approach. Our work includes the analysis of the results obtained by applying the updated Nerwip on the updated corpus, in order to design a more informed set of combination rules. Moreover, we decided to also apply an automatic approach, by training a SVM (Support Vector Machine) on our corpus.

The rest of this document is organized as follows. In section 2, we explain the concepts related to NER, and present the tools selected for this work. We also describe the updated Nerwip framework. Then, in section 3 we describe our updated corpus. In section 0, we review the existing methods used for NER evaluation. We explain their limitations and present our own method. We present and discuss the experimental results obtained by applying our platform on our corpus in section 0. Section 6 shows how we took advantage of these results to derive two different methods to combine individual NER tool outputs. Those are then tested using our corpus, and we present the resulting performances. Finally, we summarize our work in section 7, highlighting its limitations and presenting several possible perspectives.

2 NAMED ENTITY RECOGNITION

In this section, we give a definition of named entity and commonly used entity types and then discuss the methods for named entity recognition. Then we introduce the tools we selected and describe our framework, Nerwip.

2.1 Definition

A *Named Entity* (NE) is a term used to represent information units such as person, location and organization names or numeric values (e.g. date, time money and percentages). The NE term was formed during the 6th Message Understanding Conference (MUC6) (Grishman and Sundheim, 1996), which focused on Information Extraction (IE) progress. The detection of NEs in texts is an important part of IE, called *Named Entity Recognition* (NER). An entity is characterized by a *spatial dimension*, corresponding to its position in the text, and its *typical dimension*, corresponding to the kind of semantic value it represents (Person, Organization, Location, etc.). NER was widely recognized in 1996, until that point, there are several important studies for recognition of names already. After 1996, NER has become a popular domain of natural language processing, and the research community increased the amount of information in this knowledge base. Since this is a very popular research area, many tools exist to perform this task (Nadeau, 2007).

A NER tool is a system which takes a structured or unstructured text as an input and produces an output including the types and positions of detected NEs. Amongst other points, those tools differ in the processing method they rely upon, the entity types they can detect, the nature of the text they can handle, and their input/output formats. This makes it difficult for a user to select an appropriate NER tool for a specific situation.

The major purpose of this work is to detect a social network using people and events in the Wikipedia articles. To complete this objective, we need to extract spatiotemporal

entities from the unstructured texts. There are 4 major entity types for detection of spatiotemporal events: *Person*, *Location*, *Organization* and *Date* entities.

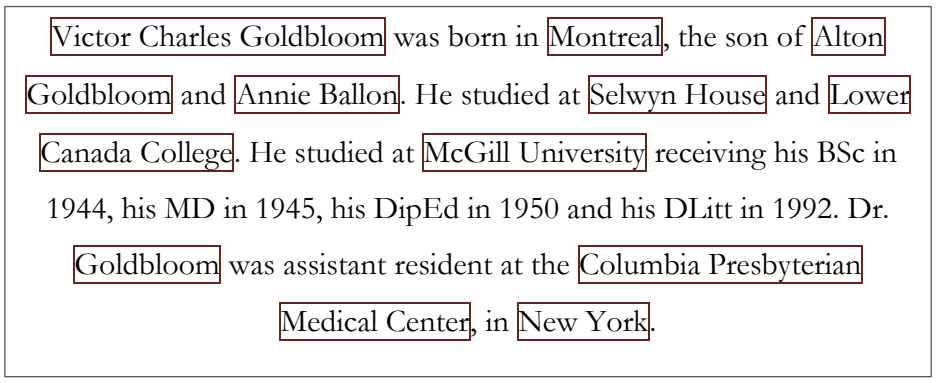

Figure 2.1. Example text for entity types

2.1.1 Person

A person name represents an individual in the text. It is usually an ordinary name, surname and sometimes including middle names. Names can have both short forms and long forms since the texts are biographies of people.

In the Figure 2.1, *Victor Charles Goldbloom*, *Alton Goldbloom*, *Annie Goldbloom*, and *Dr. Goldbloom* are person entities.

2.1.2 Location

In natural language, a location represents several different things like countries, cities, states, towns, local areas. Also organizations settled on large areas are considered as location, such as universities. In the corpus, all those levels of entities are considered as location entities, except organization names.

Location names in the example shown in Figure 2.1 are *Montreal* and *New York*.

2.1.3 Organization

In the corpus, an organization is considered as a group of people. This group may be a rules foundation like political parties, parliaments, sports teams etc.; or just a group of people like people have same political views, unions, etc.

In the Figure 2.1 organization entities are *Selwyn House*, *Lower Canada College*, *McGill University* and *Columbia Presbyterian Medical Center*.

2.1.4 Date

A date is a date written in several forms, sometimes including time information. In this work, we ignored date entities while performance assessment of NER tools phase, because not all selected NER tools support date entities.

In Figure 2.1, those are date entities: *1944*, *1945*, *1950* and *1992*. Since date entities are not considered, other date/time/datetime entity types are not shown in this example to make it simple.

2.2 Selected Tools

As mentioned before, many methods and tools were designed for named entity recognition. It is not possible to list them all here, but one can distinguish three main families (Mansouri et al., 2008): hand-made rule-based methods, machine learning-based methods, and hybrid methods. The first use manually constructed finite state patterns (Zhou and Su, 2001); the second treat NER as a classification process (Mansouri et al., 2008), and the third are a mix of those two approaches.

Amongst machine learning-based methods, three approaches are used to recognize previously unknown entities: supervised, semi-supervised unsupervised learning. Supervised learning is the dominant method for NER. The system is trained using a large annotated corpus, allowing to create a model based on the discriminative features identified in the corpus. In unsupervised learning, the corpus is not annotated which allows using a larger one (Nadeau, 2007). Semi-supervised learning is a compromise between both approaches: only a very small part of the corpus is annotated.

In order to select appropriate NER tools for our comparison, we applied the following criteria. First, the tool must be publicly and freely available. In other words, we ignored the works providing algorithms but no implementation. Moreover, there are several commercial tools with good performance, but we focused on those with a free

license for research. Second, we favored tools well known by the NER community, which generally means they have relatively good performances. Third, because the final goal of this comparison is to identify the best NER tools to extract spatiotemporal events from biographical texts, we focused on tools able to handle at least person, organization and location entities (the temporal aspect can relatively easily be dealt with in a separate tool). Fourth, we plan to work on English texts, so the tool has to handle at least this language.

In the end, we selected: Stanford Named Entity Recognizer (Finkel et al., 2005), Illinois Named Entity Tagger (Ratinov and Roth, 2009), OpenCalais Web Service (Thomson Reuters, 2008) and LingPipe (Alias-i, 2008). All of them are based on machine learning methods. Except OpenCalais, they are provided with several models. Those were trained on various corpora, and can therefore handle different text categories and entity types. These tools also allow training new models by using different corpora. In the rest of this section, we describe their main properties, in particular regarding the pre-trained models.

2.2.1 Stanford Named Entity Recognizer (SNER) (Finkel et al., 2005)

This popular Java tool is based on linear chain conditional random fields, a supervised learning method. It is provided with several predefined models for the English language. Even if it is not the case with these models, one can take advantage of dictionaries by using them during the training phase of new models.

The first model (SNER1) is based on the CoNNL03 training set (cf. section 3 for a description of the corpora), and can recognize Person, Location and Organization entities, and a generic type called Misc. The second (SNER2) was trained on the MUC6 and MUC7 corpora, and can handle seven entity types: Time, Location, Organization, Person, Money, Percent and Data. The third (SNER3) was trained on all these corpora plus ACE, and is able to recognize Person, Location and Organization entities. Each of these three models exists in a plain version and in an augmented version, which includes distributional similarity features, i.e. additional data supposed

to improve performance. Therefore, we used only the latter. These exist with or without case sensitivity.

2.2.2 Illinois Named Entity Tagger (INET) (Ratinov and Roth, 2009)

This Java tool is based on several supervised learning methods: hidden Markov models, multilayered neural networks and other statistical methods. It also uses manually annotated dictionaries for lookup, and word clusters generated from unlabeled text to improve performance. A few word clusters and dictionaries are distributed with the tool, and it is possible to build new ones. Word clusters, models, output encoding schemes can be configured via a specific file.

Table 2.1. Comparison of the tool properties

Tool	Lang.	Interface	License	Entity Types
Stanford Named Entity Recognizer (SNER)	Java	Console, Java	GPL v2	3-7
Illinois Named Entity Tagger (INET)	Java	Console, Java	Research and Academic use License	4-18
OpenCalais Web Service (OCWS)	N/A	Web API	Free API with quotas	up to 39
LingPipe (LIPI)	Java	Console, Java	Free and commercial licenses	3

The tool is provided with several models trained on English texts from the CoNLL03 corpus. As a result, they can detect Person, Organization, Location, and Misc. entities. INET allows training new ones. The first model (INET1) was generated to have a

lower bound when comparing the performances of other the configurations. The second (INET2), is the result of a single-pass process. The third (INET3) was obtained through a two-pass process; it is supposed to be better, but slower. The fourth model (INET4) is based on the same process, but it was trained on both CoNLL03 training and development sets. By comparison, the three other models rely only on the training set.

2.2.3 OpenCalais Web Service (OCWS) (Thomson Reuters, 2008)

This tool takes the unusual form of a Web service. It is free to use and have a public API for developers. However, because it is a closed source commercial product, the nature of the internal processing it performs is unknown to us, and neither is the nature of the data used for its training.

It can process English, French or Spanish raw or structured (XML/HTML) text. It supports 39 different types of entities, some of which are subsumed by the ones we target. For this reason, we associate several OCWS entity types to the same targeted type. The Person type is treated as such. A Location can be one amongst City, Continent, Country, ProvinceOrState and Region. An Organization can be of the OCWS types Company, MusicGroup or Organization. Besides NER, OCWS is able to perform other NLP-related tasks, such as detecting entity relations.

2.2.4 LingPipe (LIPI) (Alias-i, 2008)

Like OCWS, this tool is commercial and can handle various other NLP tasks besides NRE. It is open source and a free license is available for academic use. It relies on n-gram character language models, trained through hidden Markov models and conditional random field methods.

Three different models are provided for the English language. Two of them are dedicated to genetics-related texts, and are therefore of little interest to us. The third is built on the MUC6 corpus and can detect Organization, Location and Person entities. Many aspects of the process, such as the chunking method, can be controlled via a configuration file.

2.3 Nerwip

As mentioned in the introduction, the work we present here is based on the use of a platform named Nerwip (Named Entity Recognition for Wikipedia Pages). It was first developed by Y. Akbulut (Akbulut, 2010) and continued by B. Kupelioglu (KÜPELİOĞLU, 2012). The purpose of the platform is retrieving articles from Wikipedia, then applying several third party NER tools on these articles, including by combining the outputs of several NER tools, and finally assessing their performances according to various criteria.. It could process person, location, organization and misc. entities at first, then date entities were added by B. Kupelioglu. We extended it in order to support an additional NER tool, changed the evaluation measures, and developed an automatic combination method.

Inputs are processed in 5 steps: information retrieval, named entity recognition, output unification, aggregation and evaluation. In this section, we first present the general architecture of the tools, and then the details of these steps.

2.3.1 General Architecture

In Figure 2.2 we show the classes of the platform. We have 4 main classes: `SNE`, `Aggregator`, `OutputUnifier` and `Evaluator`. `SNE` is the entry point of the platform, it runs NER tools and creates entities. `Aggregator` combines all entities, then those outputs of different NER tools are normalized by the `OutputUnifier` class. All results are sent to `Evaluator` to assess the performance of the tools.

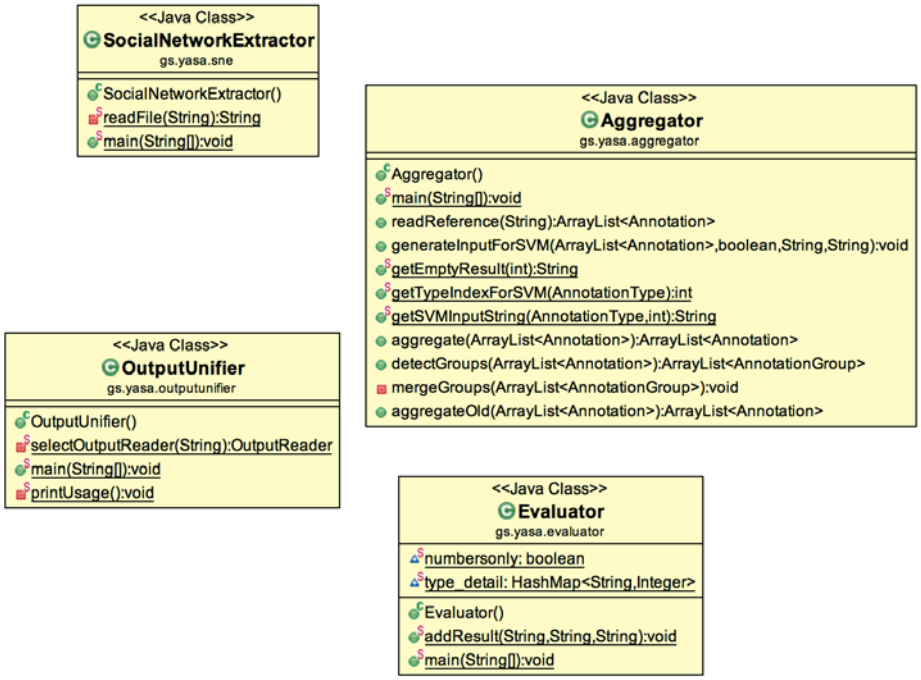

Figure 2.2. Class diagram of the main classes.

The `SNE` class contains NER tools' specific functions. To make this possible, all NER tools should implement an interface named as `NERTool`. The class diagram of NER tools and their interface is shown in Figure 2.3.

For every `OutputUnifier` instance, the platform should be aware of the output format. To do this, every `OutputUnifier` instance also has an `OutputReader` class. The base duty of these classes is reading the output with considering the format of the tool output. Since every tool has a different format, we need another `OutputReader` class. All `OutputReader` classes implement an interface whose name is `OutputReader` also. Details are shown in Figure 2.4.

We gave details of phases of Nerwip platform in subsections 2.3.2 - 2.3.6.

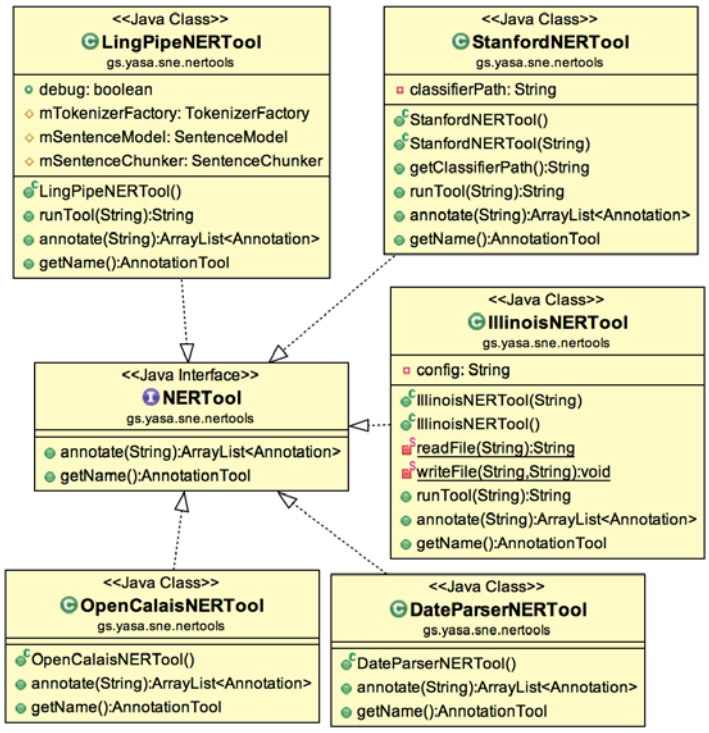

Figure 2.3. Class diagram of NER tools and `NERTool` interface

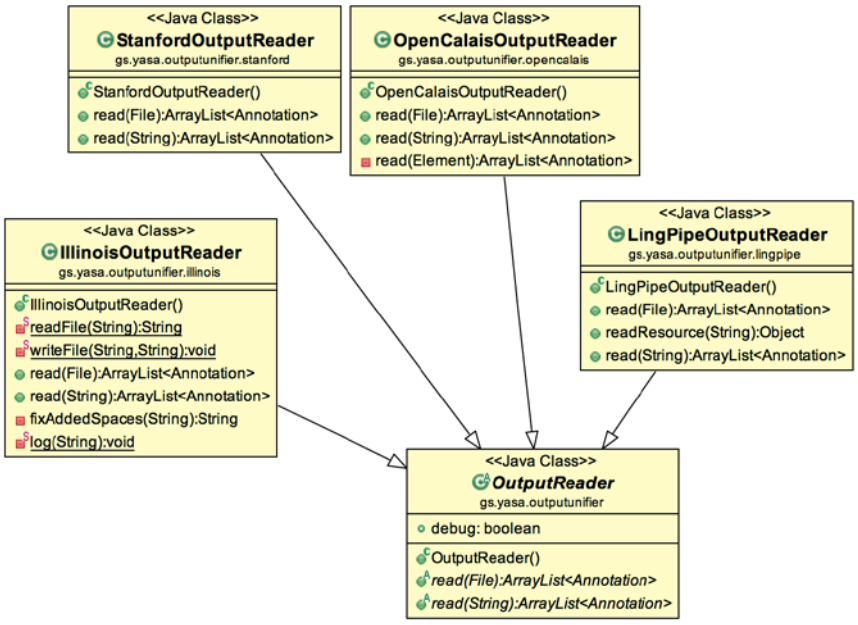

Figure 2.4. Class diagram of `OutputReader` interface implemented classes

2.3.2 Information Retrieval

In this phase, data is retrieved from the web. On the web, text data is usually unstructured or semi-structured. A Wikipedia page has a structure, the text of interest is always located at the same place. In this step, this text is retrieved and cleaned, since it is mixed with HTML and Javascript. Two different versions of the text are stored locally: the raw text, without any HTML code, and the linked text, which includes only hyperlinks.

2.3.3 Named Entity Recognition

After data is taken from the web (or from the stored files), all third party NER tools are ran over the data. In the first version of this project, there were 3 third party tools: *Stanford NER*, *Illinois NET* and *OpenCalais WS*. Also, a NER tool called *TagLinks* was developed internally and added to the platform. In the second version, a *DateParser* tool was developed and added.

In our work, we added a new third party NER tools: *Alias-i LingPipe* (Alias-i, 2008). Details concerning LingPipe are given in section 2.2.4.

2.3.4 Output Unification

There is no standard format for annotated text, so all the NER tools give different outputs after the NER phase. But we need a single format from all tools to apply our evaluation methods efficiently. For this purpose, our platform has an output unification step. It consists in creating custom Java objects for each single detected entity.
In the Java object, we store:

- Raw text
- Entity type
- Start position
- End position
- Annotation tool

These data are saved into files, for caching matters, as serialized Java objects.

2.3.5 Aggregation

In the aggregation phase, unified outputs coming from different NER tools are combined to improve the overall performance. The Aggregator is the tool taking individual NER outputs as inputs, comparing and combining them. Comparison may be done via two ways: length- or position-based. In length-based comparison, we simply compare the sizes of the detected entities, and choose the longer one. In position-based comparison, we choose the entity with greater starting position. If starting positions are equal, we choose the entity with greater ending position. These are two approaches to find the most suitable entity candidate.

In Yasa Akbulut's work (Akbulut, 2010), aggregation is based on a simple voting mechanism. Tools are given fixed weights, intuitively estimated through a limited evaluation performed on a small number of texts. Each tool votes with that fixed weight and the winner entity is determined via these votes. In our work, we followed two approaches: first we trained an SVM classifier and used the resulting model to predict the entity types. Second, as a baseline for comparison, we manually defined a set of rules based on our evaluation, of the NER tools performed on our corpus. Both approaches required a large rewriting of the Aggregator.

2.3.6 Evaluation

The Evaluator is a part of the tool taking some reference entities on one side (manual annotations from the corpus) and entities estimated by the NER tools on the other side. It compares them, and processes a series of measures to quantify their performance. Those are described in section 0. In our work, we largely modified the evaluation process developed in Yasa Akbulut's work (Akbulut, 2010). This component software was consequently mostly rewritten.

3 NER DATA

NER requires big amounts of data for both training and testing the tools. Most studies use some standard corpora generally designed for conferences or competitions, whereas some commercial tools are provided with their own data (Alias-i, 2013). These corpora are generally specialized in the sense they focus on a certain type of texts, such as news or genetics. In this section, we first review the available corpora, and show that none of them are actually appropriate for our purpose. We consequently designed our own corpus, which is presented in the second subsection.

3.1 Existing Corpora

3.1.1 NYTAC

The New York Times Annotated Corpus (Sandhaus, 2008) contains more than 1.5 million manually annotated articles published in this journal. The concerned entity types are Person, Location, Organization, Title and Topics. However, its access is conditional to the payment of a fee, and we decided to focus on freely available tools in this work.

3.1.2 MUC

The name *MUC Corpora* refers to several datasets produced for information retrieval competitions taking place in various editions of the *Message Understanding Conference* (Grishman and Sundheim, 1996). For the first five editions, the focus was on event detection, which is not exactly NER. It consists in identifying at the same time entities, and the relationships between them. Therefore, NER can be considered as a subtask of event detection, and the MUC1-5 datasets could be adapted for our use. MUC6 and MUC7 directly include data concerning NER. The treated entity types include Person, Location and Organization, but also temporal and numeric entities.

However, none of the corpora fit our needs. MUC1 and MUC5 are not publicly available; MUC2 includes only Japanese and Chinese texts; MUC3 and MUC4 focus on terrorist activity reports, which is a very particular type of text; and a fee is required to access MUC6 and MUC7 (which in addition concern only labor negotiation and corporate management, and airplane crashes and missile launches, respectively).

3.1.3 NIST IE-ER 99

This corpus was provided by the National Institute of Standards and Technology (NIST) for a competition in the information extraction domain (Doddington et al., 2004). It contains test data collected from newswires, with the following entity types: Person, Location, Organization, Date, Money, and Interval. However, it is not accessible from the web anymore, so it is not possible to use it. Moreover, it focuses on news texts.

3.1.4 CoNLL

Most editions of the *Conference on Computational Natural Language Learning* host a NLP-related competition, and provide datasets to evaluate the proposed tools. In 2002 and 2003, this shared task was NER. Both corresponding corpora are constituted of news texts, annotated using the entity types Person, Organization, Location and Misc. Texts are divided in three groups: a training set and two test sets. The first test set is meant to be used during development, whereas the second one is reserved to the final evaluation of the tool and is supposed to be more difficult to process. CoNLL02 only contains Spanish and Dutch texts (Sang, 2002) but CoNLL03 focused on the English and German languages (Sang and de Meulder, 2003). However, all the articles are related to news, not biographies. Moreover, the annotations are publicly available but their use requires to access commercial corpora.

3.1.5 Email Corpora

Four different corpora were constituted for the purpose of NER assessment in (Minkov et al., 2005). They are based on sets of emails exchanged during a Carnegie Mellon MBA class and from the Enron dataset. However, the focus is only on the retrieval of

Person entities. Moreover, the emails are not related to biographies or similar texts, but to management-related communication.

3.1.6 ACE

The ACE corpora are datasets designed by the Linguistic Data Consortium for *Automatic Content Extraction* tasks (Linguistic Data Consortium, 2005) ACE1 focuses on entities for the English language, ACE2 additionally deals with relationships, ACE2003, ACE2004 and ACE2005 extend to the Arabic and Chinese languages. The English material was collected from newswire sources, broadcast news and newspapers. ACE corpora contain Person, Organization, Location, Facility, Weapon, Vehicle and Geopolitical entity types (sometimes with subtypes). However, the access to these corpora requires the payment of a fee.

3.1.7 Summary

In Table 3.1, we summarize the main traits of the existing corpora described in this section. Our conclusion is that none of them fully fit our needs, for reasons of inappropriate category or absence of a free availability.

Table 3.1. Comparison of the corpora properties.

Corpus	Category	Access
NYTAC	News	Commercial
MUC1	Military messages	Unavailable
MUC2	Military messages	Public
MUC3	Terrorism reports	Public
MUC4	Terrorism reports	Public
MUC5	International trade	Unavailable
MUC6	Negotiations, management	Commercial
MUC7	Aeronautics, weaponry	Commercial
NIST IE-ER 99	News	Unavailable
CoNLL02	News	Commercial
CoNLL03	News	Commercial
Email Corpora	Emails	Unavailable
ACE1	News	Commercial
ACE2	News	Commercial
ACE2003	News	Commercial
ACE2004	News	Commercial
ACE2005	News	Commercial

3.2 Our Corpus

Due to the absence of a corpus meeting the needs of the our project, a new corpus was constituted by B. Kupelioglu (KÜPELİOĞLU, 2012). It was designed specifically to assess NER tool performance on biographical texts. She first extracted more than 300 biographical articles from Wikipedia, then cleaned and annotated 249 of them by hand.

Victor Charles Goldbloom was born in Montreal, the son of Alton Goldbloom and Annie Ballon. He studied at Selwyn House and Lower Canada College. He studied at McGill University receiving his BSc in 1944, his MD in 1945, his DipEd in 1950 and his DLitt in 1992. Dr. Goldbloom was assistant resident at the Columbia Presbyterian Medical Center, in New York.

Figure 3.1. Sample of a biographical text

An example text from this corpus is shown in Figure 3.1. The annotation was performed using the *Simple Manual Annotation Tool*, which is distributed with SNER. We consequently adopted its output format, which is based on an XML-like syntax, as illustrated in Figure 3.2. Since our final goal is the detection of spatiotemporal events, Person, Location, Organization and Date entities are annotated.

```
<tag name="PERSON" value="start"/>Victor Charles
Goldbloom<tag name="PERSON" value="end"/> was born
in <tag name="LOCATION" value="start"/>Montreal<tag
name="LOCATION" value="end"/>, the son of <tag
name="PERSON" value="start"/>Alton Goldbloom<tag
name="PERSON" value="end"/> and <tag name="PERSON"
value="start"/>Annie Ballon<tag name="PERSON"
value="end"/>.
```

Figure 3.2. Annotated text with XML-like tags

For each processed article, the corpus contains:

- The raw text file, to be processed by the tested NER tools;
- The linked text file, same as the raw text but with the original hyperlinks, used by TagLinks;
- The annotated version, to be used as the ground truth during evaluation.

Table 3.2. Number of articles by category

Politics	Science	Military	Art	Sports	Others
94	48	11	34	25	37

The texts concern people from six categories of interest: Politics, Science, Military, Art, Sports, and other activities (medicine, law, etc.). The distribution of articles over categories is given in Table 3.2. One person may actually belong to different categories; for example, a scientist may also be a politician. In this kind of situation, we subjectively selected the major category we thought the person belongs to. Note there are more politicians because the final goal of our spatiotemporal event extraction

project primarily concerns this population. Our selection contains people born during the 19th and 20th centuries.

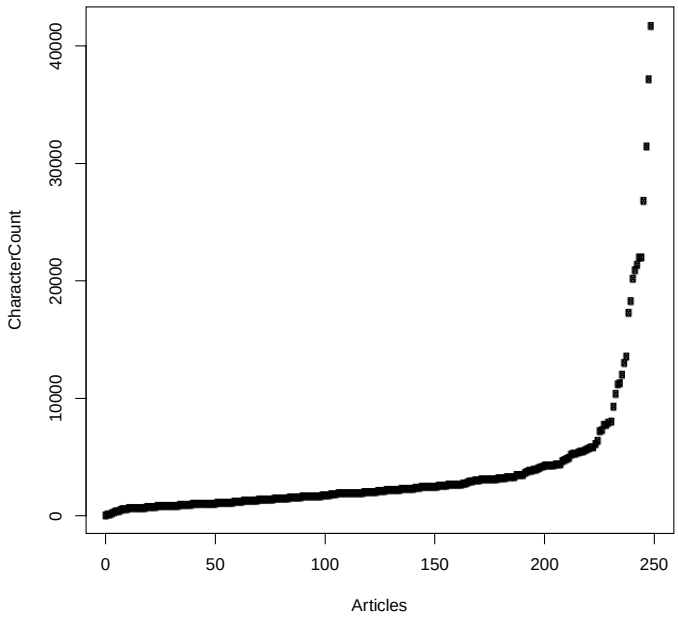

Figure 3.3. Article sizes, expressed in characters

Article length is relatively variable in the corpus: most of the biographic texts contain between 1000 and 5000 characters (including spaces), and between 100 and 400 words.

Table 3.3. Number of entities by type

Person	Location	Organization	Date
7330	2350	4611	4126

Distribution of articles by the category which category the biography belongs to is given in the Table 3.2. The corpus contains 21364 annotated entities in total. Number of entities in the corpus by entity type is shown in Table 3.3.

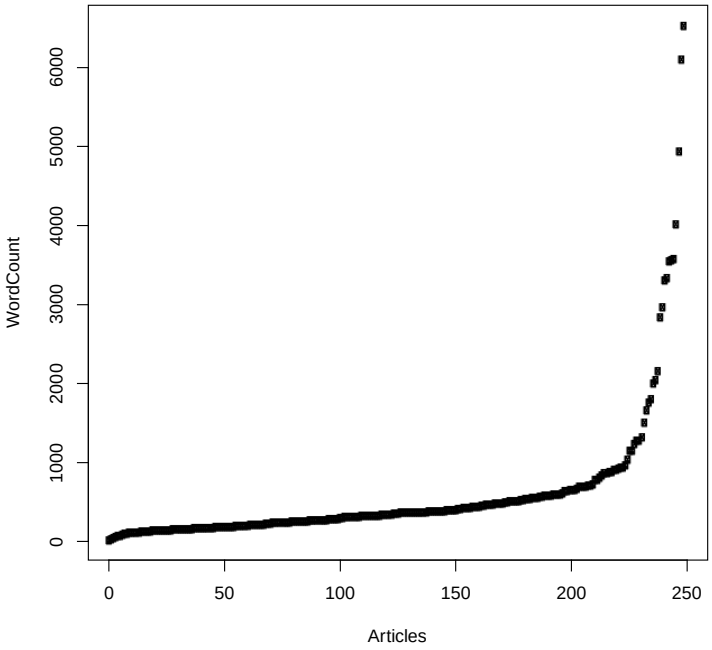

Figure 3.4. Article sizes, expressed in words

The articles were collected randomly in the various categories of interest, and interestingly, their size seems to be power-law-distributed, both in terms of characters and words. This is illustrated by Figure 3.3 and Figure 3.4, in which the articles are represented on the x axis, sorted by increasing size, and the sizes themselves are represented on the y axis.

Table 3.4. Comparison of the Corpora properties

Corpus	Size in Words	Language	Entity Types
NYTAC	N/A	English	4
MUC1	N/A	N/A	N/A
MUC2	N/A	Chinese, Japanese	10
MUC3	N/A	English	18
MUC4	N/A	English	24
MUC5	N/A	English, Japanese	47
MUC6	N/A	English	6
MUC7	N/A	English	7
NIST IE-ER 99	N/A	English	6
CoNLL02	N/A	Dutch, Spanish	4
CoNLL03	N/A	English, German	4
Email Corpora	N/A	English	1
ACE1	225 k	English	5
ACE2	270 k	English	5
ACE2003	150 k	Arabic, Chinese, English	5
ACE2004	200 k	Arabic, Chinese, English	7
ACE2005	310 k	Arabic, Chinese, English	7

4 EVALUATION METHODS

The method used to evaluate the NER task is not a trivial problem, and different approaches can be used depending on the purpose and context (Nadeau, 2007). For a given text, the output of a NER tool is a list of entities and their associated types, and the ground truth takes the exact same form. In order to assess the tool performance, one basically wants to compare both lists. In this section, we first review the traditional approaches, and then propose a variant adapted to our own context.

4.1 Classic Methods

Comparing the estimated and actual lists of entities can be performed according to two distinct axes: *spatial* and *typical*. The former refers to the boundaries of the entities in the text, and the latter to the type associated to them. A perfect detection consequently requires identifying the exact words constituting an entity, but also the appropriate type.

4.1.1 Spatial Performance

Traditionally, the focus is first on the evaluation of the spatial aspect, through specific measures. The most common ones are based on the classic *True Positive* (TP), *False Positive* (FP) and *False Negative* (FN) counts. A TP is an actual entity which was correctly detected by the tool. A FP refers to an expression considered by the tool as an entity, but which does not appear as such in the ground truth. On the opposite, a FN is an actual entity the tool was not able to detect. For matters of completeness, note that the concept of *True Negative* (TN) additionally exists in more general classification problems. It corresponds to an object which was rightfully ignored by the tool.

However, due to the overwhelming amount of such cases in the context of text mining, TN is rarely considered when assessing a tool performance.

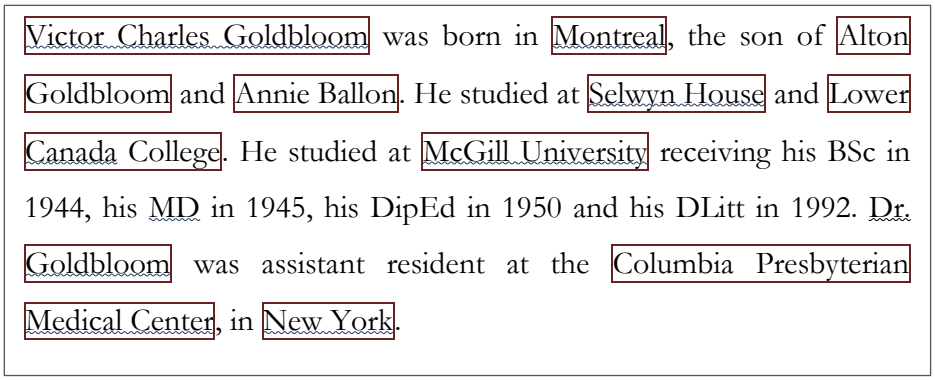

Figure 4.1. Example of annotated text

Figure 4.1 presents an example of text extracted from Wikipedia and annotated. It contains 10 actual entities represented in boxes, and 9 estimated ones characterized by wavy underlines. In terms of exact matches, there are:

- 5 TP (*Victor Charles Goldbloom*, *Montreal*, *Selwyn House*, *McGill University*, *New York*);
- 4 FP (*Canada*, *MD*, *Dr.Goldbloom*, *Medical Center*);
- 5 FN (*Alton Goldbloom*, *Annie Ballon*, *Lower Canada College*, *Goldbloom*, *Columbia Presbyterian Medical Center*).

The most widespread measures to quantify the performance of NER tools are *Precision* and *Recall*, which are based on TP, FP and FN, as follows:

$$Pre = \frac{TP}{TP + FP} \tag{4.1}$$

$$Rec = \frac{TP}{TP + FN} \tag{4.2}$$

Precision corresponds to the proportion of detected entities which are correct, whereas Recall is the proportion of real entities which were detected (Sang, 2002). Both measures are complementary, in the sense they are related to type I (false alarm) and type II (miss) errors, respectively. In the previous example, we have a Precision of 0.56 and a Recall of 0.50.

An overall measure can also be processed, under the form of the *F-measure*, which is the harmonic means of the Precision and Recall:

$$F = 2\frac{Pre \cdot Rec}{Pre + Rec} \tag{4.3}$$

In the previous example, we get a F-measure of 0.53.

4.1.2 Typical Performance

The same measures can be applied for the typical aspect of the performance, but of course with a different interpretation. TP correspond to entities whose type was correctly estimated. Due to the NER process, they consequently also correspond to entities whose position was identified at least partially correctly. FP are expressions considered by the tool as entities, but whose type was incorrectly selected, or which are not actual entities. FN are actual entities for which the tool selected the wrong type, or no type at all (Nadeau, 2007). As an example, Table 4.1 contents the types of the entity from Figure 4.1. We count 7 TP (rows 1, 2, 5, 7, 9, 10 and 11 in Table 4.1), 2 FP (rows 6 and 8) and 3 FN (rows 3, 4 and 6). Based on these counts, it is possible to process Precision, Recall, and then F-measure for the types, by using eq. (4.1), (4.2) and (4.3). For Table 4.1 example, we have a Precision of 0.78, a Recall of 0.70 and an F-measure of 0.74.

4.1.3 Overall Performance

Some authors prefer averaging spatial and typical performances to get an overall, somewhat easier to interpret, value. There are two possible ways of doing so. On the one hand, for a *macro*-averaged measure, one processes the arithmetic mean of the spatial and typical measures. On the other hand, a *micro*-averaged measure consists in summing each count (TP, FN, FP) for both space and types, and then processing the measure on these overall counts. For the example in the Table 4.1, one would get 12 TP, 6 FP and 8 FN in total, leading to a micro-averaged Precision of 0.67, a Recall of 0.60 and an F-Measure of 0.63. For the macro-averages, our example happens to result in the exact same values, which is not necessarily true in general. Finally, note there are also much more complex ways of combining both aspects of the performance (Nadeau, 2007), by assigning various weights to specific types of errors, or by considering some types as more important.

Table 4.1. Types of the entities in Figure 4.1

Reference entity	**Reference**	**Estimation**
Victor Charles Goldbloom	Person	Person
Montreal	Location	Location
Alton Goldbloom	Person	-
Annie Ballon	Person	-
Selwyn House	Organization	Organization
(Lower) Canada (College)	Organization	Location
McGill University	Organization	Organization
MD	-	Person
(Dr.) Goldbloom	Person	Person
(Col. Presb.) Medical Center	Organization	Organization
New York	Location	Location

4.2 Our Method

The previously presented measures are based on the notion of spatial match between the estimated and actual entities. In most approaches, a *complete* match is required to

count a TP: the boundaries of the estimated and actual entities must be exactly the same. However, in practice it is also possible to obtain *partial* matches (Nadeau, 2007), i.e. an estimated entity which intersects with an actual entity, but whose boundaries do not perfectly match. For example, in Figure 4.1 *Lower Canada College* is an actual entity, but the estimation only includes the word *Canada*.

However, such a match represents a significant piece of information: the NER tool detected something, even if it was not exactly the expected entity. Completely ignoring this fact seems a bit too strict to us. Moreover, in a later stage of our long-term project, we will aim at developing a method to efficiently combine the findings of several NER tools, in order to improve the overall performance. From this perspective, it is important to consider the information represented by partial matches.

We identified this point early in the long-term project, and proposed a first method to take partial matches into account. However, this method suffers from some limitations, which is why we propose a new method in this work. In the rest of this section, we first present our previous evaluation method, criticize it and present a new one, more appropriate to our needs.

4.2.1 Previous Approach

In the first version of Nerwip, as designed by Yasa Akbulut (Akbulut, 2010), a convoluted evaluation process was used. The evaluation relies on the classic TP, FP, and FN values; and on two additional values designed to assess entities detected only partially (in terms of position): *Partial Positives* (PP) and *Excess Positives* (EP). A PP corresponds to the case where the detected entity includes only a part of the actual entity. An EP reflects the situation where the detected entity includes not only the full actual entity, but also some text not present in the actual entity.

In top of this spatial aspect, the typical part of the evaluation is conducted by considering two cases in a detected entity: *correct type* and *incorrect type*. The first

case happens when the considered estimated entity at least partially matches (in terms of position) an actual entity, and moreover the estimated type and the type of said entity match. The second case corresponds to the situation where the estimated entity does not spatially match any actual entity at all, or when the matched actual entity has a different type.

Both spatial and typical aspects are combined by defining as many scores as possible situations. This results in as many as 8 measures, listed in Table 4.2. Note the distinction between correct and incorrect type is possible only when there is at least a partial match (in terms of position), which explains why FP and FN do not lead to two different type-based subcases.

Table 4.2. Evaluation measures used in the first version of Mataws (Akbulut, 2010).

Name	**Description**
True Positive & Correct Type	Exact spatial and typical match
True Positive & Incorrect Type	Exact spatial match but wrong type
Partial Positive & Correct Type	Only a part of the entity is detected, but the type is correct.
Partial Positive & Incorrect Type	Only a part of the entity is detected, and the type is wrong.
Excess Positive & Correct Type	More than the entity is detected, but the type is correct.
Excess Positive & Incorrect Type	More than the entity is detected, and the type is wrong.
False Positive	The detected entity does not match spatially any actual entity, and so there can be no typical match neither.
False Negative	An actual entity was not detected at all, so no spatial nor typical match at all.

This approach does not fit into our purposes, for three reasons. First, the output is too complicated for an efficient performance assessment, and to compare tools in an understandable way. Second, the measures are not normalized, which is a problem when comparing performances on different texts, since the magnitude of the performance values can be very different. Third (and this point also holds for class approaches described in section 4.1), the mixing of spatial and type performances prevent any fine interpretation of the results. Indeed, one of our goals with this work is to characterize the behavior of NER tools on biographical texts. To our opinion, combining the various aspects of the tool performance will result in a loss of very relevant information. To avoid this, we want to keep separated measures for space and types. To solve the problems mentioned above, we developed our approach which is presented in the following subsection.

4.2.2 Current Approach

As mentioned before, we decided to consider the spatial and typical aspects of the evaluation separately, to allow a finer assessment. For types, we decided to process Precision and Recall independently for each type. This allows assessing if the performance of a tool varies depending on the entity type. For instance, let us focus on Person entities from Table 4.1. We count 2 TP (rows 1 and 9), 1 FP (row 8) and 2 FN (rows 3 and 4). For this specific type, we therefore get a Precision of 0.67 and a Recall of 0.50.

For the spatial performance, we want to clearly distinguish partial and full matches. Not only do traditional measures consider a partial match as a no match, but they actually count it twice: once as a FN, because the actual entity is not considered to be matched by any estimated one, and once as a FP, because the estimated entity is not considered to match any actual one. Let us consider Figure 4.1 again: the actual entity *Lower Canada College* is counted as a FN, and the estimated entity *Canada* is counted as a FP. It is exactly as if there was no spatial intersection at all between the actual and

estimated entities: the result would be the same if the estimated entity was the following word, *He*.

To solve this limitation, we propose alternative counts one can substitute to the previously presented ones. First, we need to count the *Partial Matches* (PM), i.e. the cases where the estimated entity contains only a part of the actual one. We consequently also need to consider the case where the NER tool totally ignores the actual entity: we call this a *Complete Miss* (CM). The sum of PM and CM is equal to what was previously called FN. Another situation arises when the detected entities corresponds to no actual entity at all. We call this a *Wrong Hit* (WH). The sum of PM and WM is equal to FP (we remind the reader that PM are counted twice in the traditional system). Finally, the last relevant case happens when we have a *Full Match* (FM). It exactly corresponds to a FP, but we decided to use a different name to avoid any confusion. In Figure 4.1, we have 5 FM (the entities previously considered as TP), 3 PM (*Lower Canada College*, *Dr.Goldbloom*, *Columbia Presbyterian Medical Center*), 1 WH (*MD*) and 2 CM (*Alton Goldbloom* and *Annie Ballon*). These new counts are summarized in Table 4.3.

Table 4.3. Evaluation counts used in this work.

Name	**Description**
Partial Match	Estimated entity incompletely overlapping an actual entity
Full Match	Estimated entity whose positions are equal to those of an actual entity (i.e. strict TP)
Wrong Hit	Estimated entity not overlapping any actual one (i.e. FP).
Complete Miss	Actual entity is not overlapped by any estimated one (i.e. FN).

We use our new counts to adapt the Precision and Recall measures. Regarding the numerator, we now have two different possibilities: FM or PM (instead of TP). For the Precision denominator, we need the total number of *estimated* entities, which amounts to $FM + PM + WH$ (and not $TP + FP$ anymore). For the Recall denominator, we use the total number of *actual* entities, which is $FM + PM + CM$ (and not $TP + FN$ anymore). We therefore obtain two kinds of Precision, which we coin *Full Precision* and *Partial Precision*:

$$Pre_F = \frac{FM}{FM + PM + WH} \tag{4.4}$$

$$Pre_P = \frac{PM}{FM + PM + WH} \tag{4.5}$$

And two kinds of Recall, called *Full Recall* and *Partial Recall*:

$$Rec_F = \frac{FM_i}{FM + PM + CM} \tag{4.6}$$

$$Rec_P = \frac{PM}{FM + PM + CM} \tag{4.7}$$

Our measures are summarized in table Table 4.4. A *Total* Precision (resp. Recall) can be obtained by summing Full and Partial Precisions (resp. Recalls). If needed, one can then process three different F-measures, depending on whether the focus is on Full, Partial or Total measures. In the example of Figure 4.1, we get $Pre_F = 0.56$ and $Pre_P = 0.33$, so the Total Precision is 0.89. For the Recall, we have $Rec_F = 0.50$ and $Rec_P = 0.30$, resulting in a Total Recall of 0.80.

Table 4.4. Evaluation measures used in this work.

Name	**Description**
Type Precision	Precision for processed for a specific type
Type Recall	Recall processed for a specific type
Partial Precision	Precision based on spatial partial matches
Full Precision	Precision based on spatial full matches
Partial Recall	Recall based on spatial partial matches
Full Recall	Recall based on spatial full matches

5 EXPERIMENTAL COMPARISON

We applied the NER tools described in section 2.2 on our corpus from section3.2, using the measures presented in section 4.2 to assess their performance. The overall spatial counts are displayed in Table 5.1, whereas those related to types are shown in Table 5.2. The values obtained for the measures built upon those counts are displayed in Table 5.3 and Table 5.4 for spatial and typical evaluation, respectively. In order to study in details the behavior of the tested NER tools, we processed their performance not only for the whole corpus, but also by entity type and by article category.

5.1 Overall Performance

Let us first consider the overall performances. From a spatial perspective (Table 5.4), there is a clear hierarchy between the tools. When considering the total measures, i.e. the sum of full and partial measures, SNER comes second for Prediction (0.88) and first for Recall (0.93). Moreover, the part of partial matches in these results is very low. This is confirmed by Table 5.1: SNER correctly detects many more entities (FM) and misses much less entities (CM) than the other tools. LIPI has the third Precision (0.81) and the second Recall (0.89), but the part of partial matches is much higher.

INET is fourth for Precision (0.79) and third for Recall (0.78), and the share of partial matches are even more important (more than one third of the total performance). Note the fact the balance between full and partial matches changes from one tool to the other shows it is a relevant criterion for performance assessment. We manually examined the texts annotated by INET and found out this high level of partial matches has two main causes. First, many organization names include a location or a person name. INET tends to focus on them, rather than on the larger expression corresponding to the organization name. For example, in the expression *Toronto's Consulate General of the*

Netherlands, INET detects the locations (*Toronto* and *Netherlands*). Second, INET has trouble detecting person names which include more than two words.

All previous three tools reach very comparable values for both measures. However, this is not the case for OCWS. This tool has the best Precision (0.91) but by far the worst Recall (0.61), with the smallest proportions of partial matches. The unbalance between the two measures means that OCWS is almost always right when detecting an entity, but also that it misses a lot of them. This is confirmed when considering the total number of detected entities, which can be deduced from Table 5.1 by summing FM, PM and WH. We get the values 16423, 16155 and 15438 for SNER, INET and LIPI, respectively, when OCWS only detects 9530 entities (40% less).

With regards to the overall typical performances (Table 5.5), the same hierarchy emerges between the tools. SNER has the second and first Precision (0.89) and Recall (0.92), respectively. It is followed by LIPI with the third Precision (0.82) and second Recall (0.88). INET reaches the fourth Precision (0.80) and third Recall (0.78). These values mean those tools perform relatively well, and are able to appropriately classify most entities. Moreover, their performances are balanced, which is not the case of OCWS. Exactly like for the spatial evaluation, we see OCWS reach the first Precision (0.91) but the last Recall (0.61). In words, on the one hand most of the entities it recognizes are correctly classified, but on the other hand it fails to correctly classify almost half the reference entities of the corpus.

5.2 Performance by Entity Type

Let us now comment the performances by entity type. For the spatial assessment, as shown in Table 5.3, SNER performs above its overall level when dealing with Person and Location entities (especially for the former). However, its performances are under it when it comes to Organizations: full match-based measures decrease, while partial match-based ones increase. The total measures stay relatively constant, though. An analysis of the annotated texts shows SNER has some difficulties in two cases, which

mainly concern organizations. First, it tends to detect a full name followed by its abbreviation, such as in *Partido Liberación Nacional (PLN)*, as a single entity. Second, it sometimes splits names containing many words. For instance, in the phrase *Dr. Isaías Álvarez Alfaro*, it detects *Isaías* and *Álvarez Alfaro* as separate names. Finally, although it is less marked than for INET, SNER also sometimes mistakes person or location names in organization names. Regarding the typical performance, Person and Location entities are also slightly better handled: the former in terms of Precision and the latter in terms of Recall.

Concerning Person entities, the spatial performances of INET are very similar to the overall ones. For locations, the total precision decreases (due to less partial matches), whereas the recall increases (due to more full matches). In other words, INET is better as rejecting incorrect locations. For organizations, the total measures are similar to the overall level, but the share of partial matches is much higher. This means INET does not miss more Organization entities (compared to other types), but it has trouble precisely identifying their limits. In terms of typical performance, INET is clearly better on persons, both in terms of Precision and Recall. For locations, we can make the same observations than for the spatial performance, i.e. lower Precision and higher Recall compared to overall values.

For OCWS, compared to the overall results, we get similar values for locations, whereas those obtained for persons are slightly higher, and slightly lower for organizations. For all types, we observe the behavior already noticed at the overall level: Precision is high, comparable to the best other tools, whereas Recall is extremely low. A manual analysis of the annotated texts revealed OCWS has trouble handling acronyms, which mainly represent organizations in our corpus. In terms of typical performance, Person entities are also more accurately classified, and the tool is slightly better at not misclassifying organizations.

The performance of LIPI is much better on persons than overall, for both Precision and Recall: this is true for both spatial and typical measures. For locations, we observe a decrease in Full Precision and an increase in Full Recall, also for both spatial and typical results. Our interpretation is that LIPI detects more incorrect locations, but misses less correct ones. For organizations, there is a clear decrease, in terms of both Precision and Recall, with a larger part of partial matches. This last observation can be explained by the fact LIPI tends to merge consecutive organizations.

Table 5.1. Total spatial counts

	Full Matches	**Partial Matches**	**Wrong Hits**	**Complete Misses**
SNER	12511	1975	1937	1207
INET	8368	4422	3365	3801
OCWS	7506	1074	950	7517
LIPI	9752	3061	2625	2971

Table 5.2 shows distribution of correct and incorrect type detections.

Table 5.2. Correct and incorrect type counts

	True Positives	**False Positives**	**False Negatives**
SNER	13847	1791	1185
INET	12062	3134	3628
OCWS	8077	787	7314
LIPI	12240	2466	2878

5.3 Performance by Article Category

Certain article categories have an effect on the performance of certain tools. When considering SNER, there is no effect for the categories Military, Politics and Science. However, Art and Others lead to slightly lower performances, in terms of both space and types. On the contrary, the spatial performance is much higher than the overall

level for Sports (and it is also true of the typical performance, at a lesser degree). This would be due to the fact the sport-related biographies generally contain a lot of person names, such as team-mates, opponent, coaches, etc. SNER is particularly good at recognizing person names, which is why its performances are higher for this category. Art-related articles contain many titles of artworks, which are generally confusing for NER tools: they often mistake them for organization names.

Table 5.3. Spatial performance by entity type.

		Overall	Type		
			Person	Location	Organization
SNER	**FPre**	0.78	0.87	0.78	0.66
	PPre	0.10	0.05	0.06	0.19
	FRec	0.83	0.89	0.89	0.71
	PRec	0.10	0.05	0.07	0.20
INET	**FPre**	0.53	0.56	0.56	0.48
	PPre	0.26	0.28	0.13	0.32
	FRec	0.52	0.54	0.67	0.43
	PRec	0.26	0.27	0.16	0.29
OCWS	**FPre**	0.81	0.87	0.80	0.74
	PPre	0.10	0.07	0.08	0.14
	FRec	0.55	0.56	0.52	0.54
	PRec	0.06	0.04	0.05	0.10
LIPI	**FPre**	0.64	0.79	0.58	0.47
	PPre	0.17	0.11	0.14	0.29
	FRec	0.70	0.81	0.75	0.49
	PRec	0.19	0.12	0.18	0.30

For INET, we observe a clear spatial performance increase for Art articles, which means it is not concerned by the previous observation. The performance is slightly better for Science and Sports, in the sense the proportion of full matches gets higher for both Precision and Recall (the total performance staying approximately equal). On the contrary, the values are lower for Military, Politics and Sports. One difficulty with military texts is the detection of army units (e.g. *2nd Stryker Cavalry Regiment*) as

organizations. In terms of typical performance, the differences are strongly marked only for Art and Others, positively, and for Sport, negatively. So in Art articles, INET is better than usual, not only at identifying the limits of entities, but also at classifying them, whereas it is the opposite for Sport.

Table 5.4. Spatial performance by article category.

		Overall	Category					
			Art	Military	Politics	Science	Sports	Others
SNER	FPre	0.78	0.71	0.75	0.77	0.77	0.85	0.73
	PPre	0.10	0.12	0.15	0.12	0.13	0.07	0.15
	FRec	0.83	0.77	0.80	0.80	0.80	0.85	0.76
	PRec	0.10	0.13	0.16	0.12	0.13	0.08	0.15
INET	FPre	0.53	0.63	0.50	0.47	0.61	0.46	0.57
	PPre	0.26	0.17	0.24	0.33	0.21	0.26	0.26
	FRec	0.52	0.66	0.47	0.45	0.57	0.41	0.55
	PRec	0.26	0.18	0.23	0.32	0.20	0.23	0.25
OCWS	FPre	0.81	0.77	0.75	0.80	0.82	0.92	0.80
	PPre	0.10	0.11	0.16	0.10	0.12	0.05	0.12
	FRec	0.55	0.51	0.44	0.45	0.46	0.44	0.49
	PRec	0.06	0.08	0.09	0.06	0.07	0.03	0.07
LIPI	FPre	0.64	0.59	0.64	0.66	0.63	0.65	0.56
	PPre	0.17	0.19	0.20	0.18	0.20	0.24	0.25
	FRec	0.70	0.6	0.58	0.63	0.64	0.60	0.56
	PRec	0.19	0.19	0.18	0.17	0.20	0.22	0.25

In terms of spatial performance, OCWS is not very sensitive to categories: the observed performances are very similar to the overall ones. The Sports category constitutes an exception though: total Precision stays the same, but the full Precision clearly increases, meaning OCWS is able to detect entities limits more accurately. This is certainly due to the presence of more person names, as already stated for SNER: OCWS gets its best performance on this entity type. The typical performances are more contrasted. The tool is clearly better on Science articles, for which its Recall is almost at the level of the other tools (0.63). On the contrary, the Recall is very low for

Art, Others and especially Military (0.05). For the latter, it incorrectly classifies (or fail to detect) almost all the actual entities.

Like OCWS, the spatial performance of LIPI is not much affected by the article categories. For the Others category though, we observe a behavior opposite to that of OCWS for Sports: total Precision and Recall stay approximately constant, but the part of partial matches increases. It is difficult to interpret this observation, since this category corresponds to heterogeneous article themes. For the typical categories, we observe small variations. The classification is slightly better on Sports and slightly worse on Art.

5.4 General Comments

Several interesting conclusions can be drawn from our results and observations. First, even if the overall performances seem to indicate SNER as the best tool, it is difficult to rank them when considering the detailed performances. This puts in relied the fact single measures might be insufficient to properly assess the quality of NER tools and compare them. The different aspects we considered all proved to be useful to characterize the tools in a relevant way: partial matches, entity types, article categories.

Table 5.5. Typical performance by article category.

		Overall	**Category**					
			Art	**Military**	**Politics**	**Science**	**Sports**	**Others**
SNER	**Pre.**	0.88	0.83	0.91	0.89	0.90	0.93	0.87
	Rec.	0.93	0.92	0.96	0.92	0.93	0.93	0.91
INET	**Pre.**	0.80	0.80	0.81	0.80	0.82	0.73	0.83
	Rec.	0.78	0.86	0.78	0.77	0.80	0.64	0.81
OCWS	**Pre.**	0.91	0.90	0.96	0.91	0.94	0.96	0.92
	Rec.	0.61	0.34	0.05	0.53	0.63	0.51	0.19
LIPI	**Pre.**	0.82	0.78	0.85	0.84	0.83	0.89	0.81
	Rec.	0.88	0.81	0.77	0.81	0.85	0.83	0.80

As a related point, it turns out NER tools are affected by these factors in different ways. This is also why they are difficult to rank: none of them is the best on every type and category. Even some globally bad tool can be excellent in a specific context. As a consequence, these tools can be considered as complementary. For instance, if we consider types, then SNER is the best at recognizing persons. OCWS can be trusted when it recognizes locations and organizations, however is prone to missing a lot of them. On the contrary, LIPI is very good at not missing locations, but also incorrectly detect a lot of them.

The differences are not as marked for article categories, but this information can still be useful, e.g. SNER is much reliable when processing Sports articles. A natural way of taking advantage of this complementarity would be to combine the outputs of the selected NER tools, in order to improve the overall performance. This idea is explored in the next section.

6 AGGREGATION OF NER TOOLS

As shown in section 0, the tested NER tools vary in performance depending on the considered entity types and article categories. On the one hand, one tool may have a great performance on several entity types, but may be dramatically bad on some other types, or on certain categories. On the other hand, a globally bad tool may have a great performance on a specific entity type. As a result, those tools can be considered as complementary, and combining them may produce a better tool, with a better performance than all other tools taken individually.

To achieve this, we designed an aggregation process. In this section, we first, we introduce the general principle behind this process, and how we implemented it in two different ways: a classifier-based method using *Support Vector Machines* (SVM), and a vote-based one, using our observations from the previous section regarding NER tools performance. Second, we present the results obtained by applying these two approaches, to our corpus.

6.1 Aggregation Methods

In this subsection, we first present the general principle for our aggregation process. We then explain how we instantiated this principle using two different approach. The first one relies on a classifier (SVM), whereas the second is vote-based.

6.1.1 General Principle

The general aggregation process is based two phases: first a pre-processing phase, then a decision phase. In the *pre-processing* phase, we analyze the outputs of the individual NER tools, in order to identify which detected entities are likely to correspond to the same actual entity. For this purpose, we simply group together overlapping entities detected by different tools. So, concretely, at the end of this phase, we have a list of groups of entities; each group containing estimated entities supposedly representing the same actual entity. Note those estimations may vary in terms of position and type, i.e.

the NER tools do not necessarily all agree. Moreover, it is also possible that only certain tools detect an entity, so not all tools are necessarily represented in each group.

For example, let us take one location entity detected by 2 tools. Let us assume the first tool detected this entity as a location entity, and the other decided it was an organization. For other entities, one of the tools (or both) can miss detection of a particular entity, or they can false-detect a non-existing entity. Now we have two set of inputs produced by two NER tools, but we do not know if this entity belongs to location class or not.

During the *decision* phase, three choices must be made for each group detected during the pre-processing phase:

- Existence: does the group correspond to an actual entity?
- Type: if we think so, then what is the type of this entity?
- Position: and what are its exact limits in the text?

The first phase is generic, in the sense it applies as is for both methods we propose. On the opposite, the decision part varies: in the first method, we train and use an SVM classifier, whereas in the second one, we manually define a set of weights and use a vote-based approach..

6.1.2 SVM-Based Approach

The aggregation of NER tool outputs can be handled as a classification problem. For a detected entity, different tools give different (or similar) results. One can train a classifier to recognize which NER tools are generally wrong or right, depending on the general consensus and context. In our case, the classifier input correspond to the groups resulting from the pre-processing phase of the aggregation process, as presented in the previous subsection. The output of the classifier is at the same time the existence and the type of the entity. The question of its position is solved later, using an additional processing. The training data of the classifier is obtained by considering the entities defined in the reference files as the theoretical outputs.

A support vector machine is a classifier which actually acts like a decision function that accurately predicts output for unknown input, using the learning set (Isozaki and Kazawa, 2002). A SVM usually takes a set of inputs and makes a prediction about which class that inputs belong to. In Figure 6.1 x_i inputs are decisions about a type for a certain entity. The output y is a type which is decided by SVM classifier, using its trained model.

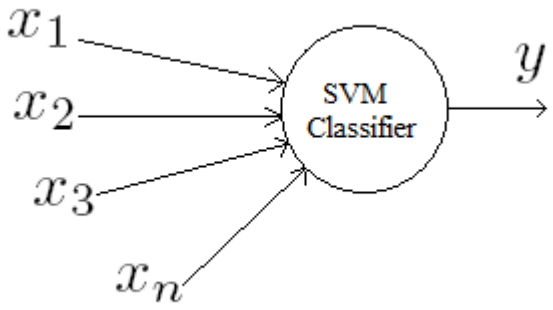

Figure 6.1. A generic Support Vector Machine

For our SVM, we need three binary inputs for each tool: onefor the PERSON type, one for the LOCATION type, and one for the ORGANIZATION type. Since we have 4 tools, it makes a total of 12 inputs. For each input, the value 0 represents the fact that *no* entity was detected, and 1 that an entity *was* detected. The output of our SVM is an integer representing either an absence of entity (value equal to 1) or the existence of an entity (value greater than 1). In the latter case, the value corresponds to the entity type: Location (2), Organization (3) or Person (4). The theoretical output is, of course, encoded in a similar way. This encoding is summarized in Table 6.1.

Table 6.1. Outputs of our SVM classifier.

Output	Estimated Type
1	No entity
2	PERSON
3	LOCATION
4	ORGANIZATION

For a detailed example, we used the same chunk of text given in Figure 4.1. Table 6.2 displays the outputs of NER tools for the example entities. The corresponding SVM inputs are shown in table Table 6.3.

Table 6.2. Outputs from NER tools and human reference

Entity number	Reference entity	Reference	SNER	INET	OCWS	LIPI
E1	Victor Charles Goldbloom	Person	Person	Person	Person	Person
E2	Montreal	Location	Location	Location	Location	Location
E3	Alton Goldbloom	Person	*none*	Person	Person	*none*
E4	Annie Ballon	Person	*none*	Person	Person	Person
E5	Selwyn House	Organization	Organization	Organization	Did not detect	Organization
E6	(Lower) Canada (College)	Organization	Location	Organization	Location	Organization
E7	McGill University	Organization	Organization	Location	Location	Location
E8	MD	*none*	Person	*none*	*none*	Person
E9	(Dr.) Goldbloom	Person	Person	Person	Location	Organization
E10	(Col. Presb.) Medical Center	Organization	Organization	Person	Person	Location
E11	New York	Location	Location	Location	Location	Location

In Table 6.1, SNER and INET detected the entity as a location, OCWS detected as an organization and LIPI could not detect it. In the reference, it is stated as a location entity.

Table 6.3. Example input for SVM

		E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
SNER	PERSON	1	0	0	0	0	0	0	1	1	0	0
	LOCATION	0	1	0	0	0	1	0	0	0	0	1
	ORGANIZATION	0	0	0	0	1	0	1	0	0	1	0
INET	PERSON	1	0	1	1	0	0	0	0	1	1	0
	LOCATION	0	1	0	0	0	0	1	0	0	0	1
	ORGANIZATION	0	0	0	0	1	1	0	0	0	0	0
OCWS	PERSON	1	0	1	1	0	0	0	0	0	1	0
	LOCATION	0	1	0	0	0	1	1	0	1	0	1
	ORGANIZATION	0	0	0	0	0	0	0	0	0	0	0
LIPI	PERSON	1	0	0	1	0	0	0	1	0	0	0
	LOCATION	0	1	0	0	0	0	1	0	0	1	1
	ORGANIZATION	0	0	0	0	1	1	0	0	2	0	0

We used libsvm for the SVM implementation. It's a well-known, widely used, good performance SVM library. Implementations are provided for several programming languages.

We divided our corpus into two parts. The first part is used for training and contains 150 articles. The second part is used for evaluation and contains the rest of the corpus, i.e. 100 articles. We first produced SVM-specific inputs using the articles from the first part. In the training phase, the SVM classifier accepts a text file containing inputs and theoretical outputs. In this file, each entity is represented as a line. A line starts with the theoretical output. Then NER tool outputs follow with a label integer. In our work, we simply used increasing numbers starting from 1. Each label is followed by the associated value. Label and information are separated with a colon sign (:). An example of input file for libsvm training is shown in Figure 6.2.

We used libsvm java classes and methods to make predictions. We simply created another dummy NER tool, which takes entities detected by other NER tools, groups the entities which belong to same entity, creates an input for each group and predicts the type using the libsvm `predict` method. The advantage of this approach is it allows applying our evaluation tools to the results produced by aggregators, too.

Theoretical Output **Features (NER tool outputs)**

```
2 1:1 2:0 3:0 4:1 5:0 6:0 7:1 8:0 9:0 10:1 11:0 12:0
2 1:0 2:1 3:0 4:0 5:1 6:0 7:1 8:0 9:0 10:1 11:0 12:0
4 1:1 2:0 3:0 4:1 5:0 6:0 7:1 8:0 9:0 10:1 11:0 12:0
```

Figure 6.2. Input format for SVM classifier training

When the training is complete, libsvm produces a model file, including all the information gained in the training phase. With this model file, predictions can be made for the new entity, whose type is unknown.

So, to summarize, the SVM classifier is in charge for the two steps of the decision phase: existence and type of the entity. The precise position of the entity in the text is determined by considering the largest entity (in terms of character count) amongst those present in the consider group of estimated entities.

6.1.3 Vote-Based Approach

As a reference for the evaluation of the SVM-based aggregator, we also manually constructed a second aggregator. For this purpose, we designed a voting mechanism, used for all three steps of the decision phase. We gave different weights to each tool, and for each of the three steps. These weights depend directly on the performances of the tools, as studied in section 0.

Let us consider the *existence* step first. All tools votes for existence of the entity. If a NER tool detects an entity, then it votes *for* the existence, and its weight corresponds to its *overall total (spatial) precision* value from section 0. If it did not detect the entity, then votes against the existence, and its weight its *overall total (spatial) recall* value. If

the total *for* weight is greater than the total *against* weight, the we conclude the entity exists. Otherwise, we consider it does not exist.

The process is similar enough for the *type* step. Of course, only the tools having voted *for* the existence of the entity can vote during this step, since the other did not estimated its type. The weight associated to each vote corresponds to the *precision* the considered too obtained for the type it detected. The type with maximal total weight is eventually selected as the final type.

During the *position* step, each tool vote independently for the starting and ending positions of the entity. Of course, like for the second step, only tools having considered the entity existed in the first step can vote. The weight associated to a vote is the *overall full (spatial) precision* previously obtained by the concerned tool. The starting and ending position with maximal total weights are selected as the final positions.

Victor Charles Goldbloom was born in Montreal.

Figure 6.3. Example sentence for rule based approach

Let us apply our rule-based approach to the example sentence given in Figure 6.3. NER tool outputs are given in Table 6.4.

Table 6.4. Annotations detected for the example sentence

NER Tool	Entity Type	Starting Position	Ending Position
SNER	PERSON	0	24
INET	PERSON	0	22
SNER	LOCATION	37	45
LIPI	ORGANIZATION	34	45

Existence step:

For the first entity, SNER and INET detects the entity, but OCWS and LIPI did not. So that, we use overall precision values of SNER and INET, overall recall values of OCWS and LIPI.

Actual entity: ***Victor Charles Goldbloom***

Precision votes = Prec(SNER) + Prec(INET) = 0.88 + 0.80 = 1.68

Recall votes = Rec(OCWS) + Rec(LIPI) = 0.61 + 0.88 = 1.49

Since precision votes are greater than recall votes, we decide this entity actually exists.

Actual entity: ***Montreal***

Precision votes = Prec(SNER) + Prec(LIPI) = 0.88 + 0.82 = 1.70

Recall votes = Rec(OCWS) + Rec(INET) = 0.78 + 0.61 = 1.39

Precision votes are greater than recall votes here too. We decide this entity actually exists.

Type step:

Actual entity: ***Victor Charles Goldbloom***

PERSON votes = Prec($SNER_{person}$) + Prec($INET_{person}$) = 0.92 + 0.84 = 1.76

LOCATION votes = None

ORGANIZATION votes = None

Since PERSON votes higher than other votes, entity type is decided as PERSON.

Actual entity: ***Montreal***

PERSON votes = None

LOCATION votes = Prec($SNER_{location}$) = 0.84

ORGANIZATION votes = Prec($LIPI_{organization}$) = 0.76

Location votes are higher for this entity. Type is detected as LOCATION.

Position Step:

Actual entity: ***Victor Charles Goldbloom***

Starting position candidates:

0 = Prec(SNER) + Prec(INET) = 0.88 + 0.79 = 1.67

Ending position candidates:

24 = Prec(SNER) = 0.88

22 = Prec(INET) = 0.79

For the starting position, we have only one candidate, 0, and it wins the voting. For the ending position, since votes of position 24 is higher than other, position 24 wins.

Actual entity: ***Montreal***

Starting position candidates:

37 = Prec(SNER) = 0.88

34 = Prec(LIPI) = 0.82

Ending position candidates:

45 = Prec(SNER) + Prec(LIPI) = 0.88 + 0.82 = 1.70

For the starting position, we have only two candidates, and votes of position 37 are higher, so it wins. For the ending position, we have only position 45 as a candidate and it wins the voting.

At the end of the three stages voting, the entities in the Table 6.5 are detected.

Table 6.5. Voting results

	Existence	Type	Starting position	Ending position
Victor Charles Goldbloom	Yes	PERSON	0	24
Montreal	Yes	LOCATION	37	45

6.2 Evaluation

We evaluated our SVM- and vote-based aggregation methods exactly like we did for the individual NER tools, in section 0: overall by entity type and by article category. This section presents these results.

6.2.1 Overall Performance

The spatial performance results are shown in the Table 6.6. For the precision, both aggregators have similar total performances. However, the proportion of full matches is higher for the rule-based aggregator, which means it is better than the SVM-based one

regarding this measure. For the total recall, the vote-based approach is clearly superior. Moreover, the proportion of partial matches is larger for the SVM-based aggregator. This means the 3rd decision step (regarding the exact position of the entities) could be improved, so that the proportion of full matches increases. Of course, this would not change the total performance, just the balance between full and partial values. So, quite surprisingly, from the spatial perspective, the vote-based approach is better than the SVM-based one. Maybe giving the SVM access to more contextual information could improve its performance. For instance, we could use the text category.

Compared to the individual NER tools, the aggregators are generally not as good as the best NER tools in terms of full matches, but they are in terms of total performance. This confirms our observation regarding possible improvement of the 3rd decision step.

Table 6.6. Spatial overall performance for aggregation and individual tools.

	FPre	**PPre**	**TPre**	**FRec**	**PRec**	**TRec**
SVM-based	0.61	0.29	0.90	0.53	0.25	0.78
Vote-Based	0.76	0.14	0.90	0.78	0.15	0.93
SNER	0.78	0.10	0.88	0.83	0.10	0.93
INET	0.53	0.26	0.79	0.52	0.26	0.78
OCWS	0.81	0.10	0.91	0.55	0.06	0.61
LIPI	0.64	0.17	0.81	0.70	0.19	0.89

Typical evaluation results are shown in Table 6.7. For the typical precision performance, both SVM-based and vote-based approaches give similar performance, but for recall value, the vote-based approach is much better. This is probably connected to its better spatial performance regarding recall, as previously observed, since undetected types count as FN.

Table 6.7. Typical overall performance for aggregation and individual tools.

	Pre.	**Rec.**
SVM-based	0.90	0.78
Vote-Based	0.90	0.93
SNER	0.88	0.93
INET	0.80	0.78
OCWS	0.91	0.61
LIPI	0.82	0.88

Compared to individual tools, both aggregator are at the level of the best one in terms of Precision, and this is also the case for the vote-based one for Recall. So, in terms of overall performance, the vote-based aggregator clearly is an improvement.

6.2.2 Performance by Entity Type

We also evaluated the aggregator performances by entity type. These results are given in Table 6.8 and Table 6.9.

In terms of spatial performance, for all three entity types, the vote-based aggregator is generally either slightly better or as good as the SVM-based one, for both Total Precision and Total Recall. Like we observed for the overall values, the difference between the aggregator is mainly the part of partial matches, which is larger in the SVM-based aggregator performance.

Table 6.8. Spatial performance for aggregation by article category

		Overall	Type		
			Person	Location	Organization
SVM-based	**FPre**	0.61	0.61	0.70	0.57
	PPre	0.29	0.32	0.18	0.24
	FRec	0.53	0.59	0.67	0.45
	PRec	0.25	0.31	0.18	0.19
Vote-based	**FPre**	0.76	0.85	0.75	0.66
	PPre	0.14	0.07	0.13	0.19
	FRec	0.78	0.88	0.83	0.71
	PRec	0.15	0.08	0.14	0.20

Regarding the typical performance, the difference is clearer: the vote-based aggregator is better (or as good) than the SVM-based one for all types, for both precision and recall.

Table 6.9. Typical performance for aggregation by entity type

		Overall	Type		
			Person	Location	Organization
SVM-based	**Pre.**	0.90	0.92	0.88	0.82
	Rec.	0.78	0.90	0.85	0.65
Vote-based	**Pre.**	0.90	0.92	0.88	0.85
	Rec.	0.93	0.96	0.97	0.91

6.2.3 Performance by Article Category

We have assessed both spatial and typical performances relatively to article categories. From the spatial perspective, displayed in Table 6.10, the vote-based agglomerator obtain better results both in terms of Precision and Recall. Like before, the total values are generally close, but the proportion of full matches is higher for this agglomerator than for the SVM-based one. In military and sports categories, the SVM-based aggregator tends to show dramatic decreases in full recall values, but we see that vote-based results has the same behavior.

Table 6.10. Spatial performance for aggregation by article category.

		Overall	Category					
			Art	Military	Politics	Science	Sports	Others
SVM-based	FPre	0.61	0.54	0.55	0.57	0.75	0.61	0.69
	PPre	0.29	0.23	0.34	0.33	0.19	0.36	0.19
	FRec	0.53	0.51	0.43	0.51	0.63	0.44	0.54
	PRec	0.25	0.22	0.27	0.30	0.16	0.26	0.15
Vote-Based	FPre	0.76	0.71	0.74	0.78	0.79	0.76	0.72
	PPre	0.14	0.14	0.17	0.12	0.14	0.18	0.18
	FRec	0.78	0.76	0.74	0.80	0.80	0.72	0.74
	PRec	0.15	0.16	0.17	0.13	0.14	0.17	0.18

In terms of typical performance, the vote-based approach dominates clearly (or is as good) in each category, as displayed in Table 6.11.

Table 6.11. Typical performance for aggregation by article category.

		Overall	Category					
			Art	Military	Politics	Science	Sports	Others
SVM-based	Pre.	0.90	0.77	0.89	0.90	0.94	0.97	0.88
	Rec.	0.78	0.73	0.71	0.82	0.79	0.70	0.69
Vote-Based	Pre.	0.90	0.85	0.91	0.91	0.93	0.94	0.90
	Rec.	0.93	0.92	0.91	0.93	0.94	0.89	0.91

In conclusion of this section, we can say the combination of several individual NER tools was successful, since it allowed to clearly improve the overall results. However, the fact that the rather raw tool we designed based on a voting mechanism was clearly better than the SVM-based aggregator trained on a part of the dataset was a surprise.

We think feeding the SVM with additional information, regarding the article context, could help improve its results.

7 CONCLUSION

In this work, we focused on the problem of selecting an appropriate Named Entity Recognition (NER) tool for biographic texts. Many NER tools exist, most of them based on generic approaches able to handle any kind of text. So, their performances on these specific data need to be compared in order to make a choice. However, existing corpora are not constituted of biographies. For this reason, we participated in the constitution of a new one, and applied a selection of publicly available NER tools on it: Stanford NER (Finkel et al., 2005), Illinois NET (Ratinov and Roth, 2009), OpenCalais WS (Thomson Reuters, 2008) and LingPipe (Alias-i, 2008). In order to highlight the importance of partial matches, we evaluated their performance using custom measures allowing to take them into account. Our results show a clear hierarchy between the tested tools: first Stanford NER, then LingPipe, Illinois NET and finally OpenCalais. The latter obtains particularly low Recall scores. When studying the detail of these performances, it turns out they are not uniform over entity types and article categories. Moreover, clear differences exist between tools in this regard. A tool like OpenCalais, which performs apparently much lower than the others (on these data), is still of interest because it can be good on niches, and therefore complete an otherwise better performing tool like Stanford NER. We decided to take advantage of this complementarity, by developing an aggregation method, able to combine the output of several NER tools. We proposed two different instances of these methods: one is SVM-based (Support Vector Machine), while the other relies on a series of voting processes, with weights determined from performances previously measured for the NER tools. Both aggregators allow improving the overall performance, and make it more stable over entity types and article categories. Surprisingly enough, the vote-based approach turns out to be better than the SVM-based one.

Our contribution includes five points. The first one is the correction of a biographic corpus previously constituted by our research group. It is based on articles of the English version of Wikipedia. A total of 247 articles were manually annotated, in order to highlight explicitly Person, Organization, Location, and Date entities. The second point is the definition of NER performance measures allowing to take partial matches into account. For this purpose, we modified the Precision, Recall and F-Measure traditionally used in text mining. The third point is the improvement of a platform allowing to benchmark NER tools, previously implemented by our research group. It is general enough to be easily extensible to other NER tools, corpora and performance measures. Our corpus and platform are both freely available online. The fourth point concerns the application of this platform to the comparison of four popular and publicly available NER tools. The fifth point concerns the definition, implementation and evaluation of an aggregation method, allowing to combine the outputs of individual NER tools, in order to improve the overall performance.

This work can be extended in several ways. First, the size of the corpus could be increased, in order to get more significant results. This would also allow using a part of the corpus for training individual NER tools, and therefore obtain classifiers possibly more adapted to process biographies than the general ones we used here. However, article annotation is a very difficult and time-costly task. Second, the benchmark could involve more NER tools, so that the results reflect more completely the possible choices of the end user. Finally, the SVM-based aggregator could be improved by using a more efficient 3rd decision step, and by considering more contextual information related to the articles, such as their category.

8 REFERENCES

AKBULUT, Y. 2010. *Extraction Automatique d'Un Réseau Social A Partir de Wikipédia*. Galatasaray University: İstanbul, Turkey.

ALIAS-I. 2008. *LingPipe 4.1.0* [Online]. Available: http://alias-i.com/lingpipe [Accessed February 22, 2013.

ALIAS-I. 2013. *LingPipe's Competition* [Online]. Alias-i LingPipe,. [Accessed 25 Mar, 2013.

DODDINGTON, G., MITCHELL, A., PRZYBOCKI, M., RAMSHAW, L., STRASSEL, S. & WEISCHEDEL, R. The Automatic Content Extraction (ACE) Program: Tasks, Data, and Evaluation. 4th International Conference on Language Resources and Evaluation, 2004 2004.

FINKEL, J. R., GRENAGER, T. & MANNING, C. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. 43rd Annual Meeting on Association for Computational Linguistics, 2005 2005. 363-370.

GRISHMAN, R. & SUNDHEIM, B. Message Understanding Conference-6: a brief history. 16th conference on Computational linguistics, 1996 1996. 466-471.

ISOZAKI, H. & KAZAWA, H. Efficient support vector classifiers for named entity recognition. 19th international conference on Computational linguistics, 2002 Taipei, Taiwan. Association for Computational Linguistics, 390–396.

KÜPELİOĞLU, H. B. 2012. *Exploitation de la Syntaxe HTML Pour la Reconnaissance D'entites Nommees.* Galatasaray University: İstanbul, Turkey.

LINGUISTIC DATA CONSORTIUM. 2005. *Automatic Content Extraction* [Online]. Available: http://projects.ldc.upenn.edu/ace/data/.

MANSOURI, A., SURIANI AFFENDEY, L. & MAMAT, A. 2008. Named Entity Recognition Approaches. *International Journal of Computer Science and Network Security*, 8**,** 339-344.

MINKOV, E., WANG, R. C. & COHEN, W. W. 2005. Extracting personal names from email: applying named entity recognition to informal text. *Conference on*

Human Language Technology and Empirical Methods in Natural Language Processing. Vancouver, Canada: Association for Computational Linguistics.

NADEAU, D. 2007. *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision.* Ottawa-Carleton Institute for Computer Science, School of Information Technology and Engineering, University of Ottawa.

RATINOV, L. & ROTH, D. Design challenges and misconceptions in named entity recognition. 13th Conference on Computational Natural Language Learning, 2009 2009. 147-155.

SANDHAUS, E. 2008. *The New York Times Annotated Corpus* [Online]. Available: http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2008T19.

SANG, E. F. T. K. Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. 6th Conference on Natural language Learning, 2002 2002. 1-4.

SANG, E. F. T. K. & DE MEULDER, F. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. 7th conference on Natural Language Learning, 2003 2003.

THOMSON REUTERS. 2008. *Calais Web Service* [Online]. Available: http://www.opencalais.com/.

ZHOU, G. D. & SU, J. Named entity recognition using an HMM-based chunk tagger. 40th Annual Meeting on ACL, 2001. Association for Computational Linguistics, 473-480.

BIOGRAPHICAL SKETCH

Samet Atdağ was born in 1984 in Adana, Turkey. He finished 19 Mayıs High School in 2002 and received his Bachelor's degree of Computer Engineering in 2011 from the İstanbul Technical University in Istanbul, Turkey. He is currently pursuing a Master's degree in Computer Engineering at the Galatasaray University, while working at Mynet as a software developer.

His first publication is a conference paper written under the supervision of the Asst. Prof. Dr. Vincent Labatut. Its title is "A Comparison of Named Entity Recognition Tools Applied to Biographical Texts", and it will be presented at the 2nd International Conference on Systems and Computer Science (ICSCS 2013) international conference.