Active Expert Sourcing

Knowledge Extraction from Domain Specific Information

Ans Alghamdi

Supervisor: Prof. Massimo Poesio
Prof. Udo Kruschwitz

Computer Science and Electronic Engineering
University of Essex

This thesis is submitted for the degree of
Doctor of Philosophy

April 2018

To my loving family . . .

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 25 figures.

Ans Alghamdi
April 2018

Acknowledgements

I would first like to thank my supervisors Prof. Massimo Poesio and Prof. Udo Kruschwitz for their support, guidance and valued advice throughout my doctoral journey. The knowledge I learned, the skills I acquired and the abilities I achieved are the things that will stay with me for the rest of my life. My thanks is extended to my examination board including Dr. Aline Villavicencio and Dr. Chris Fox from the University of Essex, Prof. Eric Atwell from the University of Leeds, Dr. Annie Louis from the University of Edinburgh, Dr. Andreas Vlachos from the University of Cambridge and the independent chair Dr. Philip Reeves from University of Essex for all of their interesting discussion and helpful suggestions. Nevertheless, I understand that by the time of producing the final version of this thesis, my supervisors left Essex. Massimo is affiliated with Queen Mary University of London and Udo is affiliated with University of Regensburg.

I would like to express my gratitude to the Albaha University, which funded my studies in Essex, as well as the School of Computer Science and Electronic Engineering for supporting my trips to conferences.

I am grateful to my collaborators from the humanities research portal the *Portale della Ricerca Umanistica in Trentino* (PRU), Francesca Bonin, Asif Ekbal, Separna Saha, Fabio Cavulli and Sara Tonelli, to Yan Yan for the help with the simulation model and to Nick Warren for technical support.

Moreover, I would like to thank the members of the research groups I have been involved with during my studies. In particular, the members of the Language and Computation (LAC) research group at the University of Essex including Jon Chamberlain, Dyaa Albakour, Deirdre Lungley, Mahmoud El-Haj, Silviu Paun and my office mate Fawaz Alarfaj. Special thanks go to Jon for reading my thesis at its very early stages. I also thank the members of the Computational Linguistics in AberdeeN (CLAN) research group at the University of Aberdeen, especially Prof. Kees van Deemter for introducing me to the group.

Finally, I want to thank my parents, Daifullah and Zaina for giving me solid foundations early on, my siblings and my wife Shorooq for supporting me emotionally.

Abstract

The development of Named Entity Recognition (NER) in recent years is partially attributed to the availability of annotated ata-sets. Data-sets play a crucial part in developing, training, and testing NER algorithms. The need for data-sets becomes more important when adapting the algorithms to new domains. However, domain specific information imposes different challenges on NERs, such as the need for annotating a different set of Named Entity (NE) types (e.g. NE schema) or, more importantly, the need for domain expert annotators. Many domain specific NER use academic paper-sharing platforms as sources for data-sets. Either abstracts or the full texts of publications are extracted from the platforms to construct raw data-sets. These raw data-sets are then annotated by domain experts. However, expert annotation is an expensive process and consumes more resources compared to non-expert annotation. This thesis tackles the problem of adapting NER to new domains and focuses on reducing the resources needed to create domain specific NER. In this thesis, academic paper-sharing portals are used as a source for raw data and also as a source for finding annotators. In other words, paper-sharing platforms are used as a crowdsourcing platform, and the scholars who share their publications are asked to annotate their own work. This thesis uses also active learning (AL) to further reduce the resources needed to develop NER. In the introduced approach, experts submit their papers online. The papers then go through a Natural Language Processing (NLP) pipeline that prepares the papers' text for annotating. An active learning algorithm, as part of this pipeline, selects the most informative instances to be annotated. The author is then asked to annotate these instances. The developed NER approach is in a consistent loop. The loop is used to produce more annotated resources and to improve the NER model. Two empirical experiments are conducted: one is a real-world experiment, and the other is a simulation. The real-world experiment tackles the archaeological domain. In this experiment, an NER is developed for two languages: English and Italian. The second experiment is in the biomedical domain, and an already annotated data-set is used to simulate the approach presented in this thesis. The results of the experiments suggest that the approach used in this thesis is a promising candidate for developing domain specific NER, as it achieved results that are significantly higher than the baseline in term of the F-score.

Table of contents

List of figures

List of tables

Nomenclature

Greek Symbols

π	$\simeq 3.14\ldots$

Acronyms / Abbreviations

HLT	Human Language Technology
NER	Named Entity Recognition
NLP	Natural Language Processing

Part I

Introduction

Chapter 1

Introduction

This chapter introduces the motivation that has stimulated this work. This section also contains a brief introduction to the area that this thesis investigates, and details the research questions that guide the development of the theoretical and empirical framework. Lastly, it lists the contribution made by this work.

1.1 Motivation

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task. It is considered to be one of the main tasks of knowledge extraction concerned with Named Entities (NE). An NE is a token in a text that can be referred to by a proper name. Therefore, NER is the process of extracting these tokens and classifying them into a specific set of types.

The majority of the work on NER deals with newswire texts. The most common types of NEs that appear in newswire texts are `Name`, `Location` and `Organisation`. Such a collection of NE types are referred to as an NE Schema, and the three aforementioned types are the `ENAMEX` schema. Newswire and `ENAMEX` became the *de facto* domain for NER and are considered to exemplify general purpose NER. Below is an example of a newswire sentence:

Henry Ford founded his car manufacturing company, Ford, in 1903.

There are four NEs in the sentence above. Henry Ford, Ford and 1903 are NEs of the `Name`, `Organisation` and `Date` types, respectively.

However, there are more uses for NERs than just newswire text. Generalising the use of an NER that was developed for newswire text to tag a text from a different genre is challenging–even if the schema has not changed. So, for example, if a data-set that contains a collection of newswire texts is used to build an NER of high accuracy, and if this NER is then used to tag NEs in emails, it will not perform as well as it does when working with newswire texts. This NER will also not perform as well when used to tag microblogs (e.g. tweets) as it does when working with newswire texts. So, either the NER needs to be adjusted to be able to tag text from other data sources, or a totally new NER must be developed.

There are also NERs that are domain-specific. These are NERs that have been trained to tag a specific text genre, or NERs that are developed to extract domain-specific NE types. There are many examples of domain-specific NERs. Some NERs are trained to extract `Ecofacts` and `Artefacts` from archaeological texts, and others are trained to extract `Drug` names from biochemical texts. Just as developing NER for a specific text genre is challenging, developing an NER for a specific schema is also challenging. The need for domain-specific NER stems from the challenges that each domain imposes.

The sample paragraph below is taken from an archaeological text [Dini and Sagramoni, 2005] and illustrates the importance of domain-specific NER. In this paragraph, a number of NEs that would be of interest to archaeologists are underlined.

> *This study about La Greppia's II US 1 lithic industry shows clearly that the site is a workshop; the abundance of debris, in particular those measuring [more] than 5 mm, the presence of, corticated flakes and blades, cores, and pre-preparation flakes of the flaking surface prove the local production of raw materials, with the identification of different knapping stages, from the preliminary knapping actions on the raw block of material to the debitage's final processes. From a typological point of view, the lithic assemblage is attributed to a transitional period between the final Epigravettian and the ancient Mesolithic (Sauveterrian). The clear predominance of backed tools, the large number of endscrapers, especially those frontal and short, and the burin's scarcity are typical of the final Epigravettian. There are also some characteristics which later become typical of the early Mesolithic, like Sauveterrian triangles and doublebacked points.*

It can be seen in the example above that the archaeology domain has a specific set of NE types that would be more interesting to extract than the general purpose `ENAMEX` types. The table below 1.1 lists some of the NEs that appear in the previous example, along with their classes. So, using a general purpose NER that has been trained on newswires would not be of much help.

Table 1.1 Some of the NEs that appear in the sample archaeological paragraph.

NE	Type
La Greppia's II US 1	Location
debris	Material
corticated flakes	Botanic Ecofact
debitage	Feature
blades	Artefact
cores	Artefact
Epigravettian	Culture
Mesolithic	Culture
Sauveterrian	Culture

There has been some work on domain-specific NER. NER has been used on almost each domain of knowledge. In the case of archaeology, there have been number of attempts to use NER. One interesting work by [Bonin et al., 2012] details how an NER for archaeological texts has been constructed. The project started by building a collaboration team between linguists, computer scientists and archaeologists. A close collaboration resulted in the development of an annotation schema containing 13 entities defined by domain experts (e.g. archaeologists). A number of published papers were used to construct a data-set. Then domain experts annotated the data-set, in accordance with the NE schema. This data-set was then used to train a supervised NER. The NER was then evaluated, and, after error analysis, the relevant steps were revised. Another interesting domain-specific NER is one used to extract Drosophila gene names [Vlachos and Gasperin, 2006]. In this work, the same method was used as in the archaeological text example. The process started by constructing a data-set of text from published articles in the domain. Then domain experts (in this case, biologists) annotated the data. Then a supervised NER was trained on the data and evaluated.

One of the challenges that domain-specific NERs have always encountered is that of constructing the initial training data-set. Constructing the initial data-set and having domain experts annotate it has proven to be the most difficult and the most resource-hungry part of developing an NER. This has sometimes involved revising NE schemas, and in some cases it has also involved re-annotating the data, which uses more resources. Since annotating data is a resource-expensive process, different methods have been presented in the literature to reduce the burden on experts. Examples of the methods used are the use of crowdsourcing, Game-With-A-Purpose (GWAP) and active learning. However, in some domains, crowdsourcing and GWAP cannot replace the contributions of experts. Therefore, a close collaboration with domain expert is the best resolution that the literature suggests.

This thesis aims to push the boundaries of knowledge extraction for domain-specific information. More specifically, this thesis describes a novel and economically viable approach to creating high-quality domain-specific NER.

1.2 Research objectives

The objectives of this thesis are as follows:

- to create an online end-to-end expert-in-the-loop framework for annotating domain-specific named entities.
- The framework developed should produce high-quality annotated data.
- The framework developed should be financially viable.
- The framework developed should help the development of domain-specific NER.
- The framework developed should make it easier to iterate the development of NER when there is a need for adjustment.
- To investigate reducing the burden on domain experts by annotating data through the use of crowdsourcing like concepts.
- To investigate reducing the burden on domain experts by annotating data through the use of active learning.

1.3 Thesis contributions

All the goals stated at the beginning of this thesis have been accomplished, and it has contributed to the field of domain-specific knowledge extraction, and specifically, the field of domain-specific NER. The list below summarises the contributions of this thesis. This thesis:

- proposes a novel method for developing domain-specific NER.
- develops a theoretical framework of using crowdsourcing experts for NER.
- translates the theoretical framework into a real-world implementation.
- empirically evaluates the theoretical framework.
- introduces the concept of using paper-sharing platforms as a method for crowdsourcing domain experts.
- suggests that there is a possibility of using the enthusiasm that scholars show when sharing their work on paper-sharing platforms. This thesis was able to get domain experts to provide annotation for the papers that they share.
- suggests that active learning can reduce the effort needed to build an NER when crowdsourcing experts.
- suggests that the framework developed offers an advantage over traditional NER development.

1.4 Thesis structure

This thesis consists of two parts: a theoretical framework and an empirical evaluation framework. In the first part, Chapter 2 surveys the landscape of NER. This chapter differs from other NER surveys in that the majority of NER surveys answer the questions How? and What? whereas this chapter answers the question Why? Also, most NER surveys focus on detailing what people use to develop NER. This type of literature is interested in how these NERs are implemented and what algorithms are used. The shortcomings in such an approach are that it does not give the reader a clear view for why there are different methods in developing NER, and it also does not tell the story behind the current status of the field. The approach taken in Chapter 2 aims to overcome such issues. It begins by defining the task of NERs, and gives examples of them, and then dives into a historical discussion of the early work on NER.

Chapter 3 talks about domain-specific NER. It presents the main challenges that domain-specific NER faces. Then two case studies of domain-specific NER are presented, one on developing NER for archaeological texts, and the other on developing NER for a very specific biological text. These two case studies have interesting details, and both case studies use active learning to reduce annotation effort, which this thesis also uses.

Chapter 4 is a survey of the field of crowdsourcing and discusses the field from a general point of view. This chapter sets the groundwork for why paper-sharing platforms have been targeted for recruiting experts.

Chapter 5 discusses active learning. This thesis uses active learning as a black box tool; therefore, this chapter does not investigate different mathematical approaches to active learning. Instead, the focus is on the use of active learning for NER. The structure of this chapter will follow a well-accepted survey of active learning by [Settles, 2012], but will give examples from the field of NER.

Then the last chapter in the first part of the thesis is the Thesis Contribution chapter (Chapter 6).

In the second part of this thesis, the experimental work starts. Chapter 7 details a real-world experiment to develop NER for the archaeological domain. It presents the framework and the method developed, then details the resources and tools used. Then the details of the different evaluation baselines and why there was a need for them are discussed, as are the number of iterations needed to develop the final version of the proposed framework.

The work in the previous chapter is then extended to a different domain in Chapter 8. However, this time a simulation experiment is conducted. The domain this time is the biological text, more specifically, the biochemistry NER is investigated. This experiment is evaluated using the same settings for the different baselines as in the Chapter 7.

The last section of this thesis (Chapter 9) presents the results and findings of the experiments and assesses the contributions of this thesis; it contains the conclusion and recommendations for future work.

Part II

Theoretical Framework

Chapter 2

Named Entity Recognition

There are a number of topics that need to be addressed to give a clear overview of the work presented in this thesis. The work in this thesis is related to developing NER for domain-specific information. Therefore, to give a clearer overview, this literature survey must cover a number of topics. Mainly, it should cover a) the definition of the NER task in general, b) domain-specific NER and c) methods for developing domain-specific NER. a) is presented in this chapter (Chapter 2) and b) is in Chapter 3, whereas c) is split into two chapters: Chapter 4 covers crowdsourcing for domain-specific NER and Chapter 5 covers active learning for domain-specific NER.

This chapter covers the history of NER. It starts with defining the task of NER and provides an example for it. Then it traces the very early work on NER. The approach taken in this chapter is quite different from any other NER survey. This chapter tries to explain why the field of NER has developed to its current status today, instead of just listing what the literature is doing. The main reason for taking this approach is that by understanding why NER has become what it has become, one can clearly see the need for domain-specific NER.

2.1 Introduction

2.2 What is NER?

The simplest way to define Named Entity Recognition is by taking the dictionary definitions [Dictionary, 2016]. Table 2.1 list the definition of each word. Putting these definitions together, NER can be defined as: the task of realising the existence of a rigid designator in text and annotating these unique identifiers of entities [Kripke, 1980] [Grishman and Sundheim, 1996b].

Table 2.1 A simple definition for named entity recognition from the dictionary.

	Definition
Named	A word or set of words by which a person or thing is known, addressed, or referred to.
Entity	Something that exists as a single and complete unit.
Recognition	The act of realising and accepting that something is true or important.

The process of extracting named entities has also been referred to by different names other than NER. One that appeared commonly in the early days is named entity recognition and classification (NERC) [Grishman and Sundheim, 1996b]. NER has also been referred to as proper name classification [McDonald, 1996], and also as

a subfield of natural language technology; more specifically, it is an information extraction (IE) task [Tjong Kim Sang and De Meulder, 2003] that aims to extract specific information from text to build knowledge.

2.2.1 Basic examples

Considering the sentence “Sir Tom McKillop left AstraZeneca in 2006, years before it moved its headquarters to Cambridge”, there are a number of rigid designators that have specific names. These items appear distinguished within the surrounding text.

The task of NER would aim to extract these rigid designators. So in this sentence, NER would be interested in the words *Tom McKillop*, *AstraZeneca* and *Cambridge*. NER would aim to classify each of these entities as `Person`, `Company` and `Place` respectively.

A more common example in literature for NER is the sentence "The automotive company created by Henry Ford in 1903 is referred to as Ford or the Ford Motor Company". This sentence includes the entities *Henry Ford*, *1903*, *Ford* and *Ford Motor Company*. The first entity is of type `Person`, the second is `Date` and the two latter ones are of type `Organization`.

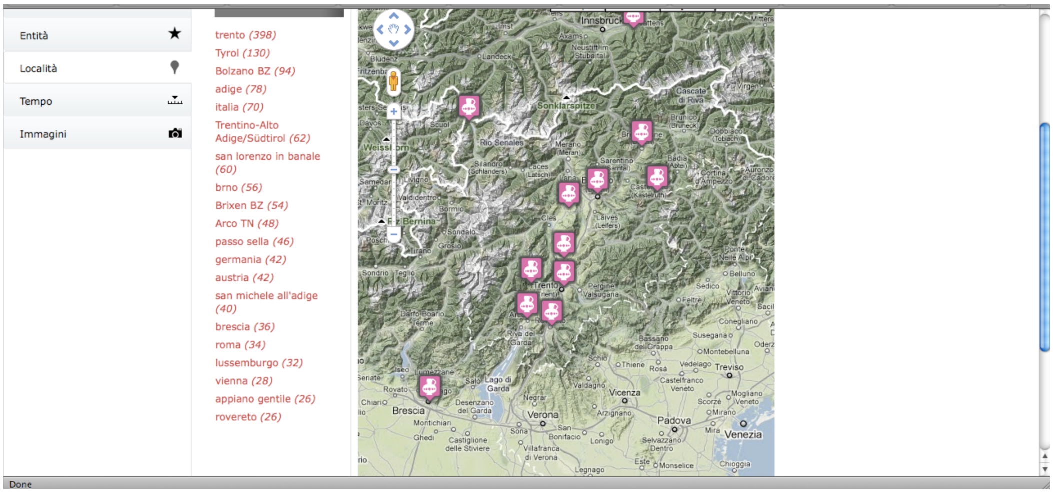
2.2.2 Why NER?

The importance of NER comes from its usefulness in many domains. Domains such as information retrieval (IR) and question answering (QaA) have found that users are more likely to search for or ask about NEs [Guo et al., 2009]. Incorporating NER in such a system tends to help users find what they are looking for. One early example of using NEs in IR that illustrates the benefit of finding NEs in queries is by factoring NEs into the joint probability $Pr(e, t, c)$ where e is an NE, t is the context and e is the class of that NE, then estimating that joint probability by applying a semi-supervised machine-learning algorithm to documents. This approach has been used in much of the literature. For instance, it has been used for offline log mining as a deterministic approach [Paşca, 2007] and for online query enhancement as a probabilistic model [Guo et al., 2009]. The utilisation of NER in search engine queries is commonly referred to as named entity recognition in query (NERQ), and is an inter-discipline involving information retrieval (IR) and natural language technology (NLP). NERQ as a field, developed in recent years from applying semi-supervised NER to logs, to applying fully supervised machine-learning NER to the whole search session [Du et al., 2010]. In recent years, NER has developed to tackle the more specialised IR problem. Examples include aiding e-commerce websites and online advertising agencies to better deliver more relevant results, by incorporating specific named entities such as `brand` and `product` names when clients search [Zhai et al., 2016]. Another example of the benefit of NER for searching in a domain-specific text is searching source code [Vinayakarao et al., 2017]. Incorporating NER in searching source code helped users, in this case developers, to use more human and natural terms when querying.

NER has also been used for QaA tasks, which tackle a similar problem to IR. However, instead of retrieving a set of relevant documents for a given query, it aims to find n possibly correct answers for the query. The Text REtrieval Conference (TREC) has had a dedicated shared task for QaA since 1999. From the very first run of a TREC QaA shared task, it has been noted that many systems tend to classify potential answers in an NER fashion [Voorhees et al., 1999], whereby a *Who* question implies looking for a `Person` NE, and a *When* question is looking for a `Date` or *Time* NE. Participants in these series of shared tasks have used different approaches over the years, however, some have utilised NER extensively to find answers [Han et al., 2004].

Nonetheless, the concept of identifying NEs in text has proven to be useful for many research fields, especially for domain-specific information. The increasing amount of information published on domain-specific paper-sharing platforms, archives and repositories has provided an incredible source for knowledge extraction [Bunescu et al., 2002]. One example is the digital humanities. The digital humanity, and, more precisely, the archaeological domain, has seen an increasing number of portals that facilitate the sharing of articles. Such portals take advantage of a number of knowledge extraction technologies, such as NER. Geographical information systems (GIS) have taken advantage of the increasing number of digitally archived historical texts [Poesio et al., 2011a]. Portals for spatial and temporal navigating through publications were made available with the help of NER. Users of such portals can explore and find publications on, for example, a particular site or a particular time period. NER in such portals identifies NEs that are of interest for this specific domain. NER has helped users not only to explore GIS portals based on sites and time frames only, but also to perform more advanced queries for artifacts or biofacts.

As seen in Figure 2.1 a GIS system has already processed articles and extracted NEs. These NEs then are used to build a map of what location each article discusses, and what artifacts were found in these locations. This is done by recognising `Artifact` NEs and linking them to `Site` and geo `Location` NEs. This map then helps archaeologists navigate though locations and find relevant information about artifacts (e.g. NEs) that they are interested in.



trento *(398)*
Tyrol *(130)*
Bolzano BZ *(94)*
adige *(78)*
italia *(70)*
Trentino-Alto Adige/Südtirol *(62)*
san lorenzo in banale *(60)*
brno *(56)*
Brixen BZ *(54)*
Arco TN *(48)*
passo sella *(46)*
germania *(42)*
austria *(42)*
san michele all'adige *(40)*
brescia *(36)*
roma *(34)*
lussemburgo *(32)*
vienna *(28)*
appiano gentile *(26)*
rovereto *(26)*

Fig. 2.1 A screen-shot of a GIS showing artifacts from a particular site. These artifacts have been annotated using NER.

However, it should be noted that publications that discuss the use of NER for archaeological texts are still not as extensive as those for the biomedical domain.

Other specialised domains that have utilised NER include, for example, the biomedical domain. The biomedical domain has extensively used NER for identifying gene and protein names [Bunescu et al., 2002]. Such advances have helped scientists search for research papers that discuss a particular protein or gene family. They have also helped business analysts find articles that mention companies in a specific industry field [Cohen and Sarawagi, 2004]. A bio-entity recognition task at the BioNLP/NLPBA 2004 is considered one of the very early attempts to tackle domain-specific NER [Kim et al., 2004]. The aim of this was to extract and correctly classify NEs in the domain of molecular biology, which is of interest to biologists. Although by the time of the bio-entity recognition task in 2004 there had been good progress in general purpose newswire NER, domain-specific NERs that were tailored to domains such as the biomedical one remained challenging [Kim et al., 2004]. Such domains pose specific challenges, such as ambiguity caused by descriptive naming or shortened forms of genes and abbreviations. However, ideas from established algorithms applied to newswire were shown to be helpful, such as using conditional random fields (CRFs), support vector machines (SVMs) and hidden Markov models (HMMs). Publications on NER for the biomedical domain have continued to increase in recent years, and an established set of tools is now available for researchers to use. Examples of such tools include ABNER [Settles, 2005], BANNER [Leaman et al., 2008] and LingPipe[Carpenter, 2007]. Nevertheless, conferences, shared tasks and publications have continued to develop this area of study until the present day.

After this brief definition of the NER task, the next section will investigate the very early work on this field of research. To give the reader a clearer understanding of this field, the sections below discuss what is commonly referred to as general purpose NER. The chapter that follows (Chapter 3) is dedicated to domain-specific NER.

2.3 Early work on NER

The very first work on NER dates back to 1991, when a system was developed to extract the names of companies [Rau, 1991]. This work describes a hand-crafted rule-based algorithm that is able to find mentions of company names in a given financial news text. This algorithm was able not only to achieve an accuracy of 95% but also was able to extract 25% more entities compared to a human annotator. The algorithm developed in this work looks for indicators of mentions of company names. The hand-crafted rules include searching for things like *Co.* and *Inc.*. The algorithm also looks for sentences with mixed-case letters, words starting with capital letters and so on. Despite how trivial this might seem, at the time when this work was published, however, it was so novel that it was developed into a fully-fledged patent [Rau, 1994].

However, it should be noted that the concept of detecting and extracting a specific piece of information that belongs to a pre-specified set of types has an older history. It is somehow related to a program started by the Defense Advanced Research Projects Agency (DARPA)[1] called the TIPSTER text program. Among the different phases TIPSTER had, was one to extract information from newswire to fill a predefined *templates*. As part of this program, a set of conferences was established for scholars to develop algorithms for extracting knowledge. The concept here is very similar to NER, which is why the community of Message Understanding Conferences (MUC), which was founded by DARPA, was able to adapt NER easily to their shared tasks.

The sixth Message Understanding Conference (MUC-6) was the first conference that had a dedicated track for NER [Sundheim, 1996]. MUC-6 had a shared task where researchers were asked to take part in a competition to develop an NER system. The task describes three groups of NEs that need to be extracted, which are:

- Named Entities (`ENAMEX`): Named entities refer to proper names, acronyms and any unique identifiers. In this group of entities, three types are to be identified:
  - `Organization`: This is any mention of a company, governmental, or other organisational entity.

[1] https://www.darpa.mil

    - `Person`: which obviously is any name of a person or family.
    - `Location`: This includes any geographically defined location, such as towns, cities, states and countries.

- Temporal Expressions (`TIMEX`): The entities to be identified under this section are temporal expressions such as `Date` and `Time` expressions. However, this subset proved to need more explanation than other types. Guidelines explained what could and what could not be counted as `Time` and `Date`. Text like "20 minutes after 10" and "midnight" were appropriate to be tagged; however, "a few minutes after the hour" and "morning" were not. `TIMEX` included some domain-specific entities, such as the expression of financial quarters (e.g. fourth quarter and first half), which are important to the business domain. Moreover, the description of this task needed to explain explicitly how to tag expressions as being multi words (e.g. tokens).

- Number Expressions (`NUMEX`): This set described the mentions of numeric expressions which include `Money` and `Percentage` expressions. This set included an explicit numeric representation and an alphabetic form.

Developing the MUC-6 shared task on NER needed the collaboration of different governmental and research bodies over one and a half years. Efforts included defining the guidelines of tagging each and every type and also having an initial dry run (e.g. anonymous) submission and evaluation. It included specific guidelines for each NE type. It also explained how to deal with the case where there are time expressions that contain another type of entity (e.g. nested expressions). An example of nested expression is "1:30 p.m. Chicago time". Here a participant needs to tag the whole expression as `Time` and also the mention *Chicago* as `Location`.

MUC-6 provided an annotated data-set for participant systems to use for development and for testing. This data-set was obtained from newswire, mainly financial news items. This reflects the fact that some of the NE types that the task asked to be annotated would appear mainly in the finance domain. Later in this section, the effect of the type of text in this data-set will be discussed.

The results of evaluating 20 systems submitted to MUC-6 reported an average of F-measure of above 90%. This result does seems high, especially given that this is the first run of such a shared task. One of the reasons for such a high score is that participants were already familiar with what NER is. Many participants were able to utilise algorithms and systems that have been already developed for similar tasks like the DARPA template-filling tasks mentioned earlier. Nevertheless, it is worth mentioning that the data-set of the MUC-6 NER task was rather simple to process, which explains the higher scores, and also was not representative of the complexity of how text appears in the real world [Sundheim, 1996]. One of the outcomes of the MUC-6 shared task was that simple NEs are simple to detect and ambiguous ones are hard to tag [Sundheim, 1996].

Participants in the MUC-6 took different approaches to NER. The sections below will detail the different approaches that different teams took. These approaches can be split into two types. The first are surface linguistic approaches, which are, in a sense, unsupervised learning NER. The other type is a probabilistic approach, which uses supervised leaning algorithms.

2.3.1 Surface linguistic approach

The very first work that stands out as an example of early NER is the one to extract company names from financial reports mentioned earlier [Rau, 1991]. From that point onward, many other works have adapted similar unsupervised learning methods. There is the LOLITA system developed by the University of Durham [Morgan et al., 1995], which is among the very first general purpose NLP systems. The development of this system started in 1986. Ten years later, LOLITA developed into NER. LOLITA mainly uses a set of complex hand-crafted rules and expressions for two tasks: information extraction and translation. Although this system originally provided some support for languages other than English, such as Italian and Chinese, for NER, it supported only English.

The development of NER in the LOLITA system is similar to Alembic, which is a system developed by MITRE [Aberdeen et al., 1995]. Both systems rely on extensive use of UNIX, C and Lisp pre-processing of the text for syntactic and semantic analysis. These systems make use of part-of-speech (POS) taggers, gazetteers, parsers and lexicon. The combination of this information extracted build-up to identify and classify NEs.

There are other systems that rely exclusively on one rule-based technique, e.g. a parser, like the Principar-driven Information Extraction (PIE) system developed by the University of Manitoba [Lin, 1995]. PIE relies on a parser named PRINCIPAR [Lin, 1993], which was developed by the same team at the University of Manitoba. However, a deeper look inside the publications that describe this system reveals that this parser is similar to the previously mentioned ones. PIE also uses information from gazetteers and lexicon analysis.

By studying a number of early works on unsupervised NER, there are three major steps such systems take when implementing NER. As in Figure 2.2 these three steps are:

- Knowledge building: This is to extract as much information from text as possible. The information extraction process includes tokenisation, POS, dictionary look-ups, gazetteers and noun phrase detection. However, this step might also involve getting information from outside sources such as Penn TreeBank [Marcus et al., 1993] and WordNet [Miller, 1995].

- NE detection: Here many systems report using one or more pattern matchings, e.g. regular expressions. This can be either a multi-layer process or just one process with a combination of pattern detectors. The aim of this step is to identify NEs that will be fed into the next step.
- Class assigning: Here systems differ greatly in how to figure out what class NEs belong to. However, the most common feature is that information gathered from step 1 is utilised to find the most likely class.

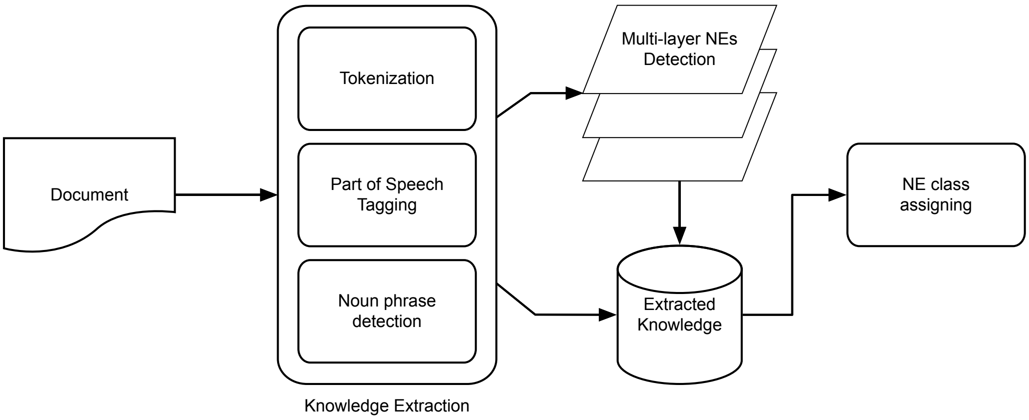


Fig. 2.2 Surface linguistic NER approaches share the steps illustrated in this diagram.

A major issue that many early systems reported was that NER needed a longer time to process and extract NEs. It was reported that processing some complex sentences can take a couple of minutes, and processing a whole document can take up to 12 minutes [Grishman, 1995]. There are other constraints that made developing NER challenging, such as having limited disk space [Baldwin et al., 1995]. However, given that these systems were being developed in the mid-90s, computational constraints are understandable.

2.3.2 Probabilistic approach

The early probabilistic approach to NER can again be credited to MUC-06. However, statistical NLPs have been an active field of research for a long time. This has resulted in the sharing of many concepts of already established statistical linguistic methods. Algorithms that once were used for parsing and POS have been successfully ported to NER.

A considerable number of participants in the MUC-06's NER shared task who used a supervised approach to NER have reported that their algorithms are part of an ongoing effort for a statistical take on NLP. For example, the probabilistic language understanding model (PLUM) system, developed by Raytheon BBN Technologies, is a machine-learning effort for NLP that started in 1991. However, in 1996, NER support was added to PLUM [Weischedel, 1995]. The details reported at MUC-06 describe how many different concepts have been borrowed from the different components of their older implementation of PLUM. The implementation of NER is based on ideas developed as part of their English parser [Magerman, 1995]. However, to support NER, the classification algorithm for NER has changed from a decision tree to hidden Markov models (HMM).

The same method of porting a statistical approach from one NLP system to NER can also be seen in the AutoLearn system developed by New Mexico State University [Cowie, 1995]. AutoLearn uses a decision tree algorithm developed for parsing [Quinlan, 1979] [Quinlan, 1991]. Another example is the UNO system by Wayne State University [Iwańska, 1995]. UNO was an ongoing effort started in 1992 for representation and inference [Iwafiska, 1992] and logical and temporal reasoning [Iwańska, 1993] [Iwarlska, 1994]. The same statistical approach is used to implement NER as part of this system.

2.4 Domain independent NER

Throughout the literature of NER, it can be noted that many NER initially were developed for some specific domain. A domain can be either a specific genre of text, a specific human language or a specific annotation schema. Nevertheless, there are attempts to generalise one NER to annotate text from different domain. This means building a language or domain independently. So, as in semi-supervised and unsupervised learning, it means an NER that can learn directly from whatever type of text is available.

Although in many cases one would aim toward building a domain-independent NER, however, some bias can still be found. Many developed algorithms for NER tend to be biased toward the domain that it initially developed on. This bias can either be intentional, like in designing an NER for a specific domain, or unintentional, such as lacking access to more representative training data.

A paper describing the design and evaluation of the MUC-6 NER shared task back in 1996 stated that the task aimed to develop domain-independent technologies [Grishman and Sundheim, 1996a]. However, after the conference tracks were run, there were a number of observations made by the organisers [Sundheim, 1996] in regard to domain-independence. These observations can be summarised as the following:

- Text bias: the text represents only one style of writing, which is journalistic text. This is because the data-set provided for participants at MUC-6 is a subset of the Wall Street Journal corpus (which is available at the Linguistic Data Consortium (LDC)).

- Language bias: the whole shared task was NER for the English language. This might be because it was the first run of such a shared task. Therefore, different languages have been introduced in the following years' shared tasks series.

- Schema bias: the NEs that participants were asked to look for are more related to financial text than any other text genera. MUC-6 described three main schemas for MEs: proper names (`ENAMEX`), temporal (`TIMEX`) and numerical (`NUMEX`) expressions. `ENAMEX` can, to some extent, be considered domain-independent set of types. However, `TIMEX` and `NUMEX` are more financial newswire related NE types. Schema bias is related to text bias, as the type of text can determine what type of NEs to look for. So in the case of schema bias and text bias, one could be the result of the other.

The issue of bias and domain-dependent NER is a longstanding problem that many researchers have tried to investigate. Domain related issues in NER can be divided into three categories. The following sections shed some light on the issues presented in the literature in this regard.

2.4.1 NER for specific text

The issue of text bias can be traced to the first NER shared task of MUC-6. The data-set that was used for developing and testing participating systems was a subset of a financial newswire corpus. This is not a negative or bad practice, in fact, when it comes to the MUC series of conferences, this bias is introduced intentionally. A close look at other earlier MUC shared tasks reveals that they are related to NER to some extent. Participants were asked to fill in a specific *template*, with a specific set of information from a given text.

In the first set of MUC conferences, each task was aimed toward a specific genre of text. MUC-1 was an exploratory shared task where participants were asked to extract information, with no strict rules. The text provided was part of a fleet of operations reports, which involves information from naval sightings and engagement forms of military messages. The same data-set was used for MUC-2, but giving participants more specifications for what to extract[2]. MUC-3, on the other hand, shifted the type of text in the shared data-set from navy traffic messages (operational reports) to newswire. The focus of MUC-3 and MUC-4 was still military oriented, where the goal was to extract information from reports of terrorist events in Central and South America. The data-set shared as part of MUC-3 was a collection obtained from an electronic database containing articles gathered by the Foreign Broadcast Information Service from a number of worldwide open source news outlets. The data-set, although it was newswire, was still military oriented. The focus of MUC shifted, however, this time, from military related newswire to business related newswire in the MUC-5 version. This answers the question why the MUC-6 shared task was to extract NEs from business related newswire.

The shared tasks of MUC conferences can demonstrate how text bias affects machine learning in general and NER in particular. A paper published at MUC-3 [Hirschman, 1991] explains in detail this effect, with a number of observations. What is relevant from these observations to the discussion in this section is what is called *complexity of the data*. There are many factors that can have an input into the complexity of a specific text, for example, the domain where the text comes from. Some text genre can have a fairly limited vocabulary.

There is also the issue of punctuation and run-on sentences that differ from one domain to the other, not to mention that for some domains, as in the operational report of the US Navy, text tends to have some sort of telegraphic syntax. Newswire, on the other hand, has a more standard syntax, and tends to cover a wider range of topics. Newswire also has a somewhat larger vocabulary set. Other factors that make one genre of text differ from another include:

[2] During the production of this thesis, it was not possible to find publicly accessible records of MUC-1 and MUC-2 proceedings. Therefore, information given here is referenced from what is reported in this regard from subsequent MUC series proceedings.

- Rates of how often new words appear
- Different types of grammar used
- Sentence and word patterns

There is other literature that discusses the issue of text bias. Some studies examine the effect of author, audience and formality of a text in NER [Maynard et al., 2001]. Such studies have suggested that there are differences between data-sets that are constructed from written, spoken or email languages, even if the domain is not changed. An example of this is that when a data-set is constructed of text from emails, the text tends to be less formal, contain spelling mistakes and does not follow grammar and capitalisation standards. Table 2.2 summarises some of the differences between data-sets of different resources. So when a system is trained on one type of text (e.g. newswire) and then is applied to extract NEs from a different type of text (e.g. emails), it has been found that it performs differently, if not poorly. This observation still happens even if both texts are from the same knowledge domain (e.g. business related topics).

Table 2.2 Some of the differences in data-sets constructed from different resources, where ✓ means mostly correct or less error, and ✗ means missing or high error rate.

	Written	Spoken	Email
Spelling	✓	✓	✗
Punctuation	✓	✗	✗
Capitalization	✓	✓	✗
Spacing	✓	✗	✓
End of lines	✓	✗	✗

One interesting case of NER for specific text has appeared in the biomedical domain. Machine-learning NER in the biomedical domain has been applied successfully. ML algorithms such as conditional random fields (CRF) and the hidden Markov model (HMM) are among the mostly utilised algorithms for NER, not only in the biomedical domain, but ML NLP in general. Such systems depend on the availability of data-sets for training. One source for training data for such domain-specific NLP tends to be abstracts of scientific papers published online. So, in the biomedical domain case, abstracts of articles from the PubMed repositories tend to construct most data-sets. CRF-based NER has been applied to the biomedical domain [Vlachos, 2007b], as well as HMM-based NER [Vlachos and Gasperin, 2006]. In both cases, the data-sets used for training were obtained from abstracts of scientific papers. The interesting observation is that when these systems tried to annotate the full text of papers, both performances dropped dramatically [Vlachos, 2007a]. So, in this case, the text from abstracts happened to have different characteristics than full papers text –even though domain, language, source and writing style seems to not be the same.

2.4.2 NER for specific language

The language factor in the NER literature is clearly presented. English being the dominant language for NER, much research has been devoted to other languages. The literature spectrum in this regard has covered almost every spoken language, starting with the MUC-6, when the Japanese language was part of the shared task of that conference [Grishman and Sundheim, 1996a].

However, the issue of NER for specific languages received more attention in the 2002 and 2003 versions of the SIGNLL[3] Conference on Computational Natural Language Learning (CoNLL) conferences [Tjong Kim Sang and De Meulder, 2003][Cucerzan and Yarowsky, 1999]. In this conference, shared tasks on NER were introduced, where participants were asked to develop a language-independent NER. In the 2002 version, two languages were tackled: Spanish and Dutch, and in the 2003 version it was English and German. These conferences, and many other subsequent ones, advertise shared

[3] SIGNLL, pronounced as "signal", stands for Special Interest Group on Natural Language Learning, of the Association for Computational Linguistics (ACL).

tasks as language-independent. However, the reality is that participants are actually developing systems for a particular set of languages.

The issue of language-independent (or NER for specific languages) is actually an inherent issue. Organisers of shared tasks provide annotated resources in a particular language, yet ask participants to develop a language-independent algorithm. When participants take part in such shared tasks, no matter how language-independent their techniques are, they are still bound to evaluate their techniques on specific predefined languages. To tackle the issue, conferences have long allowed participants to utilise any outside data-sets or resources, and use not only the data provided by the organisers. The hope is that one day an algorithm can learn NER, then extract NEs from any text, regardless of the language.

A number of attempts have appeared in the literature to tackle the issue of NER for a specific language. Attempts span from the MUC shared tasks in the 1990s and the CoNLL shared tasks in the 2000s, to the most recently shared tasks such as entity discovery and linking (EDL) at TAC2015 and TAC2016 [Ji et al., 2016, 2015]. A different take on this matter appeared in one of the workshops of the 15th conference of the European Chapter of the Association for Computational Linguistics (EACL 2017). A challenge was organised as part of the 6th Workshop on Balto-Slavic Natural Language Processing (BSNLP 2017). BSNLP 2017 organised a shared task on multilingual NLP: the First Cross-Lingual Challenge on Recognition, Normalization and Matching of Named Entities in Slavic Languages [Piskorski et al., 2017]. In this shared task, no training data was given to participants, although evaluation was on specific predefined languages. So, similar to previous shared tasks where participants were allowed to utilise other resources that were not provided by the organisers, in this case, participants were forced to figure out what resources to use all by themselves.

2.4.3 NER for specific schema

Researchers tend to specify a schema for NEs, as NEs that appear in one domain may not appear in another. For example the ENAMEX collection of NE types includes Person, Organization and Location. The TIMEX schema includes Time and Date, and the NUMEX schema includes Money and Percentage expressions.

The schema ENAMEX is one of the very popular collections of types that much NER literature utilises. Well-known shared tasks such as MUC and CoNLL and many more have used it extensively. Newswire, being a popular text genre, led in one way or another to the popularity of ENAMEX. NEs such as Person, Organization and Location appear in newswire frequently, and the schema ENAMEX in such cases are reasonable to apply. From time to time, organised shared tasks have introduced changes to widely used schemas (e.g. ENAMEX). For example, CoNLL-2013 [Tjong Kim Sang and De Meulder, 2003] introduced Miscellaneous, which encompasses all other NE types that are not in ENAMEX. In fact Miscellaneous, proved to be a good fallback class for any other NE that a schema does not include, and has been used in all CoNLL subsequent shared tasks on NER.

ENAMEX is considered the *de facto* schema for NER. Much research published in the field of NER uses this as a starting point for its approach. It therefore can be considered as a general purpose coarse-grained schema [Fleischman, 2001]. There are many derivatives of ENAMEX that aim for fine-grained schemas. Fine-grained ENAMEX schemas can be gained though introducing subtypes of Person, Organization and Location. Fine-grained derivative schemas from ENAMEX, TIMEX and NUMEX in the literature are expanding, especially after the 2004 Automatic Content Extraction (ACE) Program [Doddington et al., 2004]. This conference introduced a task similar to the MUC NER shared tasks, however, there are some fundamental differences. There was a broader definition of what counted as an entity, in the sense that anything that can be referred to as entity must be tagged. This includes detecting descriptions of entities and even pronouns, which, according to the organisers, makes the task about detecting "things that aren't there." ACE introduced nine more NE types, including `Facility`, `Weapon`, `Vehicle`, `Government`, `Commercial`, `Educational` and `Non-profit`; the last four types are subtypes of the class geo-political entity (GPEs).

Throughout the literature, the number of subtypes has been growing from eight [Fleischman, 2001] to nine to 10 [Tanev and Magnini, 2006] to 12 [Hovy et al., 2006] to 18 [Weischedel and Brunstein, 2005]. There are more extreme numbers of subtypes, like 112 [Ling and Weld, 2012] and 147 [Lee et al., 2007] types of NEs. Table 2.3 shows some common subtypes of the widely used coarse-grained NER schemas. Moreover, these extreme numbers of classes are, in some cases, the result of the close relationship between the NER literature and the word sense disambiguation (WSD) literature. Using NE types and subtypes is common practice for WSD, and, in some cases, the number of subtypes can even reach the staggering amount of over 8,000 subclasses [Bunescu and Pasca, 2006].

2.5 Summary

At the start of this chapter, a discussion on the definition of the NER task was established, followed by examples illustrating the usefulness of this task. The examples given range from domains related to NLP, like IR, to examples from other domains, like the archaeology and biology domains. This chapter tried to trace the early work in the field of NER. The reason for this approach is to investigative why newswire become the dominant domain for NER. Through this investigation, a number of observations were found, such as the bias that most general purpose NER have. Most general purpose NERs tend to perform well in one particular domain, despite the many efforts to build domain-independent NER. This bias can be towards language, schema or domain.

One take from the investigation on this chapter is that if there is no escape from domain bias, then this issue should be taken as a feature. This means that there is a need for domain-specific NER: NER that is tailored to tackle a specific domain's problems. Domains such as archaeology will benefit more from NER that is built exclusively to address their needs. As discussed in this chapter, the benefit is likely to be higher than using general-purpose NER (e.g. ones trained on newswire for `ENAMEX`).

The following chapter will address the need for domain-specific NER. It will discuss two cases of domain-specific NER, namely, NER for archaeological text and NER for biological text.

Table 2.3 Widely used general purpose NE subtypes.

ENAMEX			TIMEX	NUMEX	Miscellaneous	
Artifacts	Organization	Location	Date	Quantity	Study Field	Plant
Music	Economy	Others	Duration	Age	Science	Fruit
Theatricals	Education	Country	Day	Size	Social Science	Flower
Art Craft	Military	Province	Month	Length	Medicine	Tree
Cultural Asset	Media	County	Year	Count	Art	Grass
Religion	Sports	City	Season	Man Count	Philosophy	Type
Building	Art	Capitalcity	Geoage	Weight	Animal	Part
Philosophy	Society	River	Dynasty	Percentage	Others	Material
Musical Instrument	Medicine	Ocean	Time	Speed	Insect	Element
Road	Religion	Bay	Minute	Temperature	Bird	Metal
Weapon	Science	Mountain	Duration	Volume	Amphibia	Rock
Transport	Business	Topography	Second	Order	Reptilia	Chemical
Works	Library	Continent	Hour	Price	Type	Liquid
Geography	Law	Tour	Event	Phone	Fish	Gas
Medical Science	Politics	Space	Sports		Part	Term
Art	Person		Activity		Mammalia	Color
Dance	Name		Festival			Direction
Movie	Myth		War Revolution			Climate
Literature						Shape

Chapter 3

Domain Specific Named Entity Recognition

In the previous chapter (Chapter 2), the field of NER was discussed. That chapter addressed the question of why the field of NER has become what it is today. A number of issues were identified in the history of this field. These issues resulted in the need for domain-specific NER.

This chapter studies the need for domain-specific NER. It will start by illustrating examples of some specific cases of NER. A number of observations from these examples will be discussed. This discussion leads to the identification of the main challenges faced by domain-specific NER. Following that, two case studies of domain-specific NER are discussed. In these cases, a number of problems are identified; these problems are to be investigated in the experimentation chapters.

Following this chapter, two surveys are presented. The use of crowdsourcing for NER is discussed in Chapter 4, and the use of active learning to bootstrap NER is discussed in Chapter 5.

3.1 Introduction

Developing NER for domain-specific information involves a number of challenges, such as the digital challenges faced by humanity. These challenges include the exposure of tools to different text genre, different document structure and different vocabulary. Different domains expose other challenges. Thus, developing any new NER for any new domain requires addressing each domain's specific needs.

3.2 Challenges

To illustrate some of the challenges of knowledge extraction for domain-specific information, two case studies are presented. In these case studies, a thorough investigation is done to identify issues and ways to tackle them. Moreover, a number of possible areas that can be improved upon are identified. This thesis is built on these areas. This thesis presents a number of novelties that push state-of-the-art methods for domain-specific NER.

3.3 Approaches

This section presents a number of examples of domain-specific NER. The approaches used in these examples are presented. These approaches include unsupervised learning, semi-supervised learning, crowdsourcing and active learning.

3.4 Case study 1: NER for archaeological text

Digital libraries have benefited greatly from the knowledge-extraction methods that have been applied to publicly available repositories. Repositories such as the ACL[1]

[1] http://aclweb.org/anthology/

allowed researchers to access various information related to articles—such as the article's citation, as well as the authors' names and affiliations. This allowed services such as Google Scholar to get information from the citations to build the authors' profiles.

Moreover, interest in the extraction of NEs from articles in digital libraries has increased. It allowed users to perform deeper searches for articles and entities of interest. It allowed not only the linking of NEs in one article but also cross-article linking. Such methods allowed the users to find articles that discuss any particular topic based on NEs.

NLP has played a great role in the digital humanities knowledge extraction. It allowed knowledge extraction in published papers and provided different presentations for the papers. For instance, the use of NER allowed archaeological digital repositories to use spatial browsing for geotagging. NER and NLP in general technologies allowed users to browse the articles based not only on the article's date of publication but the temporal metadata extracted from within the text. This allows users to search for text about a specific historical period of time. This type of search is of greater help in the digital humanities than it is in other fields such as engineering.

However, archaeological repositories have faced some challenges. The main challenges can be summarised in this list:

- Portals that facilitate access to articles are fairly limited.
- Articles in the archaeological domain tend to include texts written in more than one language.
- Each institute in the world has a small set of articles.
- The number of articles that are available in digital format is limited; most articles are still in paper form.
- NE types that appear in archaeological text are not of a standard nature.

The Humanities Research Portal or The *Portale della Ricerca Umanistica in Trentino*[2] (PRU) is an effort to create a portal to facilitate searches for articles on subjects related to the humanities. The Anonymous Archaeology Lab developed it to allow the visualisation of scholarly publications in the Alps. The navigation is facilitated through the provision of a geographical information system (GIS). The portal mainly focuses on articles published in the archaeological domain. The content of this portal is in both English and Italian. The portal utilises a number of knowledge-extraction methods that extract metadata from articles published on this portal. The metadata extracted include the following:

- The main language of the document
- Document structure
- The language of each part of the document
- Archaeological named entities (e.g. temporal and spatial NEs)
- The document's citation
- The article's authors and their affiliations

The extraction of this information allows scholars to navigate the collection of publications based on the mentions of particular sites or locations. It also allows users to browse a map of areas of interest and then find articles that mention these areas. The portal is built to allow two types of searches: spatial and temporal searches.

The use of NER in this portal (and the use of different NLP methods) allows more advanced searches. For example, one can search for articles that discuss *shellfish* in a particular *site*. This kind of search means that, irrespective of whether "shellfish" appears in the text as *Spondylus sp.* or *Spondilo*, it still can be found. Regarding the

[2] http://apsat.mpasol.it/

temporal browsing of articles, one can find articles that discuss a particular time period, in addition to articles published within a particular time period. For this reason, the PRU uses an automated pipeline that processes articles as they are uploaded. Various elements, such as the title, authors, abstract, keywords, semantic metadata and citation, are automatically extracted.

To give more details on how an archaeological NER was developed, the next section will detail the pipeline used as part of the PRU. The following section will shed the light on the various pieces of information extracted by the PRU pipeline.

3.4.1 Knowledge extraction

The PRU includes several knowledge-extraction stages. These stages are implemented in a pipeline. Each article that is selected to be digitalised goes through this pipeline. The knowledge-extraction pipeline of the PRU contains a number of modules that interact with each other, as shown in Figure 3.1.

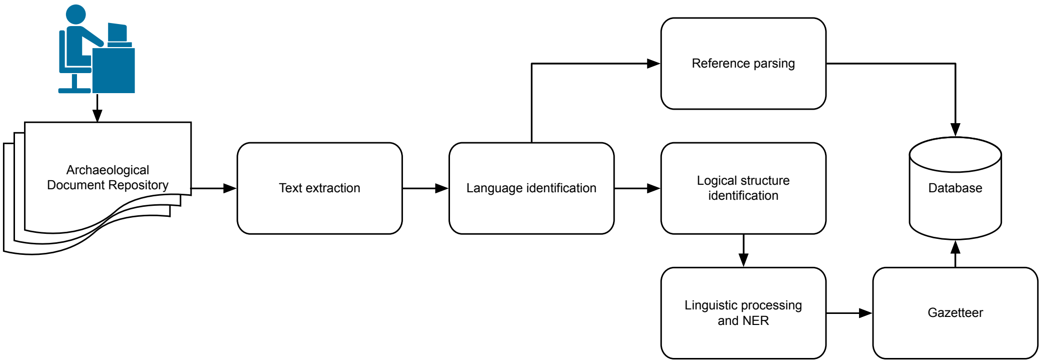

Reference parsing
Archaeological Document Repository
Text extraction
Language identification
Logical structure identification
Database
Linguistic processing and NER
Gazetteer

Fig. 3.1 The architecture of the PRU pipeline.

These modules can be summarised in the following list:

1. **Text extraction.** All the articles in the collection are available in PDF format, and no `txt` version is available for them. Therefore, it is necessary to convert files from PDF to text format or extract the text from the PDFs. Various open-source

software have been tested, and the one the pipeline uses is *pdftotext*. This software support uses a variety of encodings, which is essential for the PRU, as the articles to be processed are written in different languages. However, it was noticed that this software sometimes does not preserve the text's original order.

2. **Language Identification**. As mentioned earlier, the documents to be processed are multilingual. In fact, one challenging aspect is that one document might contain more than one language. For example, one article might have its abstract and conclusion in Italian, whereas the remainder of the document is written in English. The software that the pipeline has implemented is *TextCat* language guesser.

3. **Structure extraction**. Preserving the structure of the articles is an important part of the PRU. This process includes identifying different sections of the articles, such as the title, abstract, authors, affiliations and main body of the article. The archaeological papers tend to have a different nature than other published papers. Therefore, an off-the-shelf software cannot be used on them. The archaeological articles tend to have many figures. They are also written in different languages, and some documents contain more than one language. The text that is to be processed is not perfect, due to the issues mentioned earlier regarding the extraction of text from PDF files. Therefore, a custom version of *ParsCit* was trained on annotated archaeological articles. In fact, two models were trained. One was trained on articles written in English with little Italian text, and one was trained on articles written mainly in Italian with some English sections.

4. **NLP**. A number of NLP tasks—such as POS, stemming, lemmatisation and NER—are performed on the different extracted sections of the articles. Each of these tasks adapts to the language of the relevant section. Thus, if a section is

identified as written in English, then an English POS is used. On the other hand, when a section is written in Italian, an Italian version is used.

5. **Reference parsing**. The aforementioned software *ParsCit*, which extracts the structure of the papers, is used here. In a previous step, this software extracted the sections that are to be processed in this step. The extracted citation is then processed to correct possible errors. The list of citations is then stored, along with the other information. The step here prevents duplications in the stored citations, as every extracted citation is checked against the available record.

What can be noticed from the list of pipeline modules is that the pipeline relies on available and open-source software. This allows a reduction in the resources needed to develop the pipeline. The reliance on open-source tools might be the result of insufficient funding in domains such as archaeology, as sufficient funding would allow the development of tools that are tailored toward the needs of such domains.

3.4.2 Constructing data-sets

Now that a pipeline has been developed, the next step is to construct a data-set. The data-set here is needed to train the NER module so that the module is able to extract more NEs from the resources that are uploaded to the PRU.

The data-set used to develop NER needs to be split into three sub-sets of data, as illustrated in Figure 3.2. One part is for the development phase of the NER. This data-set is used to build features so that the NER can learn from it. Then, once specific features are established, the NER will use the second data-set.

The second data-set is called the training data-set. The training data-set is the data-set that the NER uses to learn a model. This model is what the NER is going to use to annotate new, unseen text. Because this model is going to annotate new data, it needs to be highly accurate.

To assess the accuracy of the NER model, the third data-set, which is called the testing data-set, is used.

The testing data-set is used to measure the accuracy of the developed NER. The NER will annotate the testing data-set without seeing the “true” annotation. The testing data-set is sometimes referred to as the “gold standard” in literature. Annotating the test data-set results in two sets of annotations: one is part of the original data-set (the gold standard), and the other is the new annotation produced by the model. Then these two sets of annotations are compared to check the model’s accuracy.

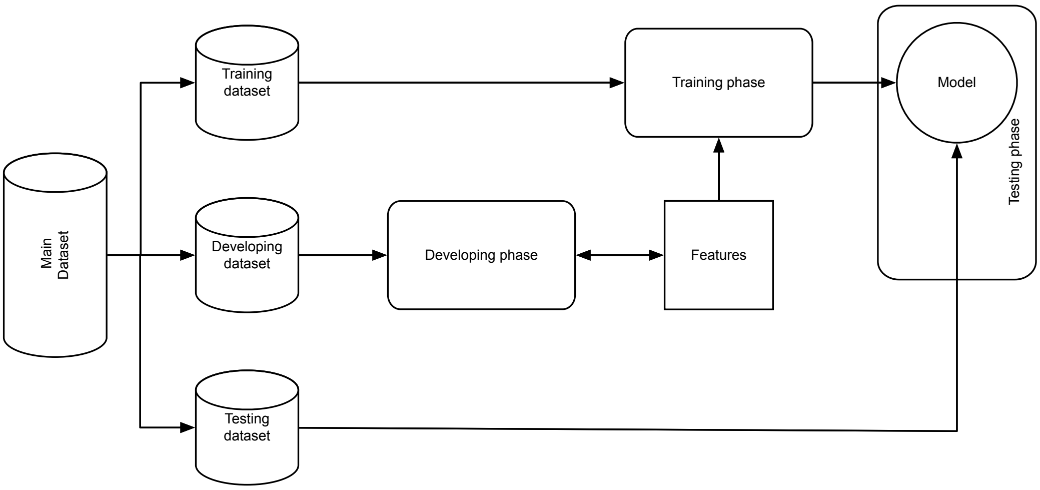


Fig. 3.2 Data-sets splitting for developing NER

The method described above is mostly used in supervised learning, in which an algorithm learns from some available data. This algorithm is then applied to extract new knowledge from unseen data. The data-set that was used in the case of the PRU is a collection of text from academic published articles. This approach is very common in many NLP tasks in which data-sets are constructed of text from publications (e.g. papers, journals). The PRU project had access to a collection of articles from the journal Preistoria Alpina, which is published by the Museo Tridentino di Scienze Naturali [Bonin et al., 2012]. This collection consists of papers written in English and Italian. The full-text files of a number of English and Italian papers were used to construct data-sets. Two parts of the PRU processing pipeline use the supervised method. One of these parts is the document-structure-extracting module. The other is the NER module. In the following sections, the process of how these data-sets were built is discussed.

3.4.3 Document structure data-set

The document-structure-extracting module relies on a *ParsCit*, a software that extracts the structure of the document by using the supervised method. Thus, this software would need a data-set to learn from. This data-set needs to have been already annotated with gold labels.

The PRU project built a data-set of 55 documents. The data-set consists of 35 Italian documents and 20 English ones. The structure of each document has been extracted and represented in XML format. Table 3.1 details the sections that were annotated. Then, this data-set will be used to learn *ParsCit*.

As mentioned earlier, the data-set needs to be split into three parts. Each part is used for its aforementioned purpose. Last, the testing set is used to measure the accuracy of the document-structure-extracting module. Table 3.2 shows the accuracy results that were detailed, based on the parts of the document. It can be noticed that some parts of the document are easier to guess, whereas other parts seem to be hard to guess.

The process of annotating the document-structure-extracting data-set was deemed resource consuming [Poesio et al., 2011c]. Annotators had to read each paper to annotate each section, which is a time-consuming process. This annotation process was only used in the document-structure-extracting module. However, another module also needs an annotated data-set: the NER module. Thus, the creators of the PRU pipeline needed to come up with a better solution to facilitate the annotation of data. A more

Table 3.1 The number structure tags (annotation) for the PRU document structure extraction module

Section	Number of tags
ItalianFigureCaption	456
ItalianBodyText	347
EnglishFigureCaption	313
SectionHeader	248
EnglishTableCaption	58
ItalianTableCaption	58
Author	71
AuthorEmail	71
AuthorAddress	65
SubsectionHeader	50
VolumeInfo	57
Bibliography	55
EnglishSummary	31
ItalianKeywords	35
EnglishKeywords	35
Title	55
ItalianSummary	29
ItalianAbstract	10
Table	25
EnglishAbstract	13
Note	18

efficient solution needed to be implemented. The solution was to use an approach called *active learning*. The questions of what this approach is and how it was used to build the NER module are discussed in the next section.

3.4.4 NER data-set

The development of a supervised NER needs annotated data. The data-set that is to be built for the PRU's NER module is, as mentioned earlier, a collection of text from published papers. Numerous papers were selected for inclusion in this data-set and therefore needed to be annotated.

Table 3.2 The result of the evaluation of the PRU's document-structure-extracting module

Section	F1
ItalianFigureCaption	70
ItalianBodyText	90
EnglishFigureCaption	71
SectionHeader	90
EnglishTableCaption	70
ItalianTableCaption	75
Author	72
AuthorEmail	75
AuthorAddress	73
SubsectionHeader	65
VolumeInfo	85
Bibliography	98
EnglishSummary	40
ItalianKeywords	55
EnglishKeywords	56
Title	73
ItalianSummary	40
ItalianAbstract	50
Table	67
EnglishAbstract	50
Note	70

The process of annotating data for the document-structure-extracting module was resource consuming. Thus, a better solution needed to be implemented to reduce the effort needed for annotation. Therefore, active learning was used.

However, before talking about active learning, the following section will discuss the annotation schema that has been developed as part of the PRU project. It is important to mention that the annotation schema here shows how domain-specific NER has a fairly different set of characteristics and how different sets of NEs only appear in archaeological text. This is why active learning was needed to reduce the resources needed to annotate data-sets.

Nevertheless, active learning will only be discussed briefly in this chapter, as a thorough review of it is provided in Chapter 5.

3.4.5 Annotation schema

One of the PRU's main purposes is to allow for entity-based search and navigation. It needed to meet the very specific needs of the archaeology domain. The portal allows users to navigate based on the number of entities that appear in the publication. Therefore, published articles have to be processed and indexed based on the named entities they contain.

Indexing papers based on general-purpose NE schemas, such as `ENAMEX`, will not provide much help for the PRU. Therefore, it is important to develop a domain-specific NE schema for archaeology papers. The PRU investigated a number of schemas but, in the end, built its own schema [Bonin et al., 2012]. The developed schema was done in close collaboration with humanities scholars. The annotation schema is shown in table 3.3.

3.4.6 NER development

The development of the NER module as part of the PRU took many cycles. In each cycle, error is analysed to improve the quality of the NER. The annotation schema was also revised to help the NER module perform better. Relevant observations from the development of domain-specific NERs are as follows:

- Data-set is constructed of published papers.
- General-purpose NE types such as `Name` and `Time` caused some confusion for the classifier. The annotation schema had to be revised to fine grain these NEs.

Table 3.3 Annotation Schema for Named Entities in the Archaeology Domain

NE Type	Details
Culture	Artefact assemblage characterising a group of people in a specific time and place
Site	Place where the remains of human activity are found (e.g. settlements, infrastructures)
Location	Geographical reference
Artefact	Objects created or modified by men (e.g. tools, vessels, ornaments)
Material	Found materials (e.g. steel)
AnimalEcofact	Animal remain different from artefacts but are still culturally relevant
BotanicEcofact	Botanical remains (e.g. trees and plants)
Features	Remains of construction- or maintenance-related dwelling activities (e.g. fireplaces, postholes)
ProposedTime	Dates that refer to a range of years hypothesised from remains
AbsTime	Exact date given by a C-14 analysis
HistoricalTime	Macro period of time referring to time ranges in a particular area
Pubyear	Publication year
Person	Human being, discussed in the text (e.g. Otzi the Iceman, Pliny the Elder, Caesar)
Pubauthor	Author in bibliographic references
Researcher	Scientist working on similar topics or persons involved in a finding
Publoc	Publication location
Puborg	Publisher
Organisation	Association (no publications)

- Although the initial manually annotated data-set gained a high inter-annotator agreement (a kappa value of 0.8), the result of the NER was not as wanted.

- The revised annotation resulted in a kappa value of 0.85, a higher inter-annotator agreement score. The NER results were also higher.

The aforementioned details about the initial NER were done in a traditional way. The data-set was annotated, then used to train a model. This is illustrated in Figure 3.2. The model was tested, and if it needed improvement, the cycle was performed again. Analysis of the model could reveal the need to revise NE schemas.

Nevertheless, the developers of the NER module took a different approach to bootstrap their NERs, whereby, active learning was utilised. Active learning is a method that allows a machine-learning algorithm to determine what to learn from. It allows a machine-learning algorithm to select instances it thinks would be more beneficial if labelled. It then asks an expert to label these instances (Figure 3.3). A more thorough discussion of active learning is provided in Chapter 5.

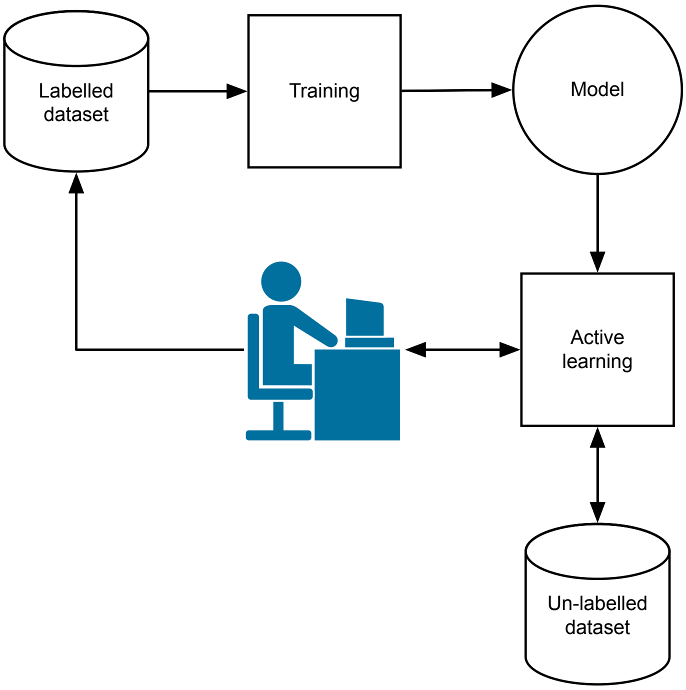


Fig. 3.3 Active learning cycle

In the PRU project, active learning was used for two things: a) to reduce the effort experts needed to annotate data and b) to increase the performance of the NER.

Reducing effort

Experts, archaeologists in this case, were needed to annotate data-sets. Then, after training and evaluating the NER, revising annotation was needed. Therefore, experts were needed again to re-annotate the data. Many domain-specific NERs face this obstacle. Experts are needed to constantly improve the performance of the NER and must re-annotate the data when there is any change in the schema.

Increasing performance

More annotated resources do not necessarily translate to better performance. Therefore, the limited resources of domain-specific NERs should be utilised wisely. Active learning can make better decisions on what to annotate and what not to annotate. By using an active-learning algorithm, whenever the algorithm needs help learning or improving, it asks an expert for help. Therefore, instead of annotating a whole data-set, training the NER model then testing performance, a small set of annotated data is used to bootstrap the active-learning NER algorithm. This algorithm then makes decisions on what needs to be annotated. This method increases the performance of the NER with fewer annotated resources.

An earlier work for *Drosophila* gene name recognition [Vlachos and Gasperin, 2006] inspired the use of active learning to create an NER pipeline for domain-specific information in this case study (the archaeological text). This earlier work formed the next case study for domain-specific NERs.

3.5 Case study 2: NERs for biomedical text

Facilitating access to scientific literature is important for many knowledge domains, and annotating the information in published literature to create data-sets helps navigate through them. However, the process to curate and annotate data-sets, especially for domain-specific information, is expensive, where domain experts are needed to curate and annotate data.

Manually annotated data-sets are required for knowledge extraction and information-extraction systems using a supervised machine-learning algorithm. In the case study of biomedical NERs, there was a need to create a system that helped navigate literature based on the NEs that appear in them. The main focus was on the names of the Drosophila gene as part of the FlyBase project.

The very first attempts to create NER gene names in FlyBase was done using a Hidden Markov Model (HMM) algorithm [Morgan et al., 2004]. The same method was extended to use more data and revise annotating guidelines [Vlachos and Gasperin, 2006]. However, in both cases, it was reported that annotation was expensive, and a need to come up with a long-term approach to the development of the IE system emerged. Other observations are summarised in the following list:

- Testing the same algorithm on two different testing data-sets resulted in significantly different results.
- Annotation guidelines needed to be revised to achieve better results. Therefore, instead of having only one NE type, `gene name`, there were three types: `gene name`, `gene mention` and `other mention`. For example, a sentence that used to be annotated as

  $the < gn > faf < /gn > gene$

  is then tagged as

  $< gm > the < gn > faf < /gn > gene < /gm >$

- Two annotators handled the manual annotation of the data-set. However, only one of them is a domain expert. The other is a computational linguist.
- The computational linguist found it very difficult to annotate, so he focused mainly on identifying the boundaries of noun phrases.
- When one system that achieved an F-score of 81.5 was ported to a slightly different domain, it performed badly. It only achieved 36.7 in terms of the

F-score, even though the task was still identifying gene names but for a different biomedical genre.

3.5.1 Improving biomedical NERs

The challenge of this case study is the development of NERs integrating into the FlyBase system. Domain experts who are already working on the FlyBase project do most of their work on a data-curating system. In this data-curating system, domain experts view documents that are to be included in the FlyBase databases and annotate these document with relevant information, thus maximising the gain from those experts where needed.

As part of the number of experimentations on the FlyBase biomedical NER, an active-learning algorithm was implemented. Different algorithms were used to train the NER [Vlachos and Gasperin, 2006] [Vlachos, 2007a] [Vlachos et al., 2006]. This is similar to the previously mentioned case study on digital humanities (e.g. archaeological text). But, to be more precise, the previous case study is actually similar to this one. The following section will shed light on the active-learning method that was used in this case study.

3.5.2 Active learning

The use of active learning in the biomedical domain (e.g. FlyBase) is due to the lack of training data. The developed NER for FlyBase [Vlachos and Gasperin, 2006] [Vlachos et al., 2006] is mainly a semi-supervised NER. This illustrates the need for a fully supervised NER that shows better results in other domains.

The main idea is to use an unsupervised algorithm to produce an imperfect training data-set, then have domain experts detect and annotate errors in this data-set. As Algorithm 1 shows, the initial step is to have a set of labelled data L and a set of unlabelled data U. Then, an NER A is trained on the labelled data-set to build model M. This model is used for two things: a) to annotate the unlabelled data-set L and b) to help a query module q to select instances that, if labelled, will improve performance. The selected instances are then shown to an annotator or expert who will label them. Lastly, the newly annotated instances are removed from the unlabelled data-set U and added with tier annotation to the labelled data-set L. This process is looped until there are no more unlabelled data or until it is interrupted and stopped.

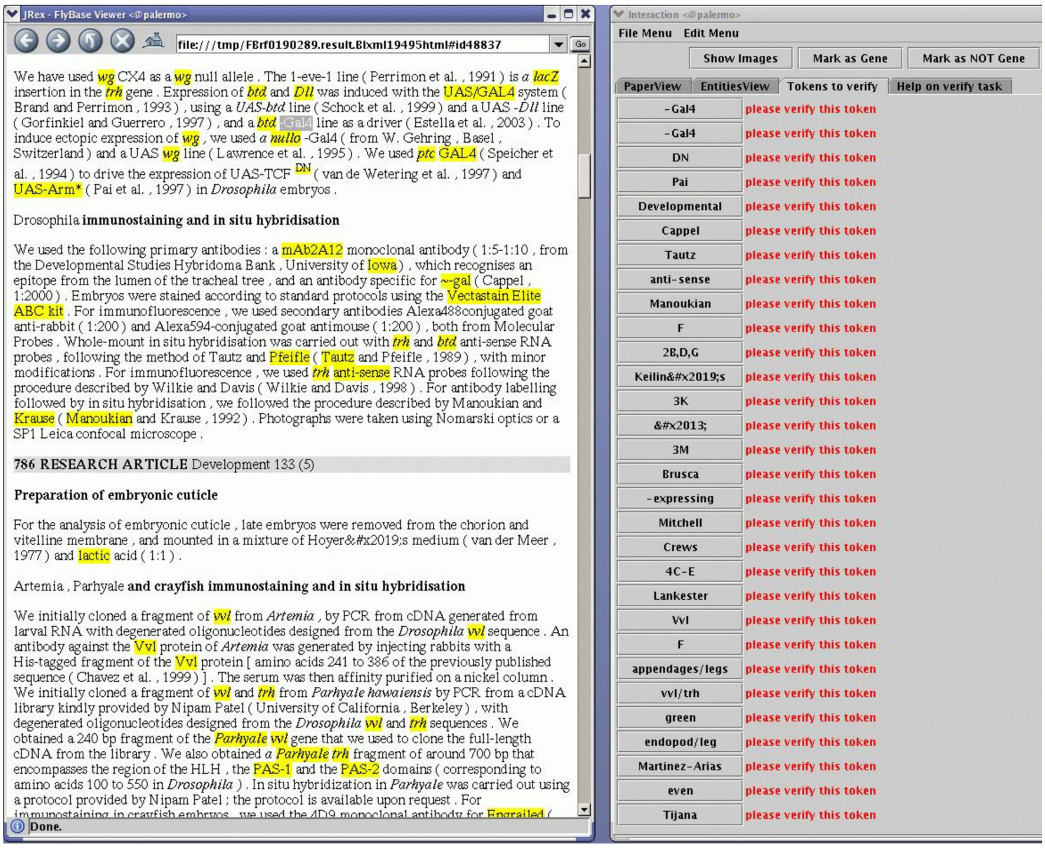
Algorithm 1 Active learning algorithm

```
$A$ is a learner.
$M^i$ is a model of $A$ at step $i$.
$U$ is a pool of unlabelled examples.
$L$ is a manually labelled seed data.
$q$ is a query module.
**Initialization**
$M^0 \leftarrow Train(A, L^0)$
**repeat**
    $N \leftarrow$ Select $n$ examples using $M^i$ according to a query method $q$.
    $U \leftarrow U - N$
    $L^{i+1} \leftarrow L^i \cup Label(N)$
    $M^{i+1} \leftarrow Train(A, L^{i+1})$
**until** $(U = \phi)$ or (human stops)
```

The algorithm described in Algorithm 1 is then integrated into the FlyBase data-curating system.

3.5.3 Integrating AL to FlyBase

After the AL algorithm in the previous section is integrated into the FlyBase curating system (Figure 3.4, experts can now provide feedback on items that the NER is uncertain about.



We have used *wg* CX4 as a *wg* null allele . The 1-eve-1 line (Perrimon et al. , 1991) is *a lacZ* insertion in the *trh* gene . Expression of *btd* and *Dll* was induced with the UAS/GAL4 system (Brand and Perrimon , 1993) , using a *UAS-btd* line (Schock et al. , 1999) and a UAS -*Dll* line (Gorfinkiel and Guerrero , 1997) , and a *btd* -Gal4 line as a driver (Estella et al. , 2003) . To induce ectopic expression of *wg* , we used *a nullo* -Gal4 (from W. Gehring , Basel , Switzerland) and a UAS *wg* line (Lawrence et al. , 1995) . We used *ptc* GAL4 (Speicher et al. , 1994) to drive the expression of UAS-TCF ^{DN} (van de Wetering et al. , 1997) and UAS-Arm* (Pai et al. , 1997) in *Drosophila* embryos .

Drosophila **immunostaining and in situ hybridisation**

We used the following primary antibodies : a mAb2A12 monoclonal antibody (1:5-1:10 , from the Developmental Studies Hybridoma Bank , University of Iowa) , which recognises an epitope from the lumen of the tracheal tree , and an antibody specific for ~-gal (Cappel , 1:2000) . Embryos were stained according to standard protocols using the Vectastain Elite ABC kit . For immunofluorescence , we used secondary antibodies Alexa488conjugated goat anti-rabbit (1:200) and Alexa594-conjugated goat antimouse (1:200) , both from Molecular Probes . Whole-mount in situ hybridisation was carried out with *trh* and *btd* anti-sense RNA probes , following the method of Tautz and Pfeifle (Tautz and Pfeifle , 1989) , with minor modifications . For immunofluorescence , we used *trh* anti-sense RNA probes following the procedure described by Wilkie and Davis (Wilkie and Davis , 1998) . For antibody labelling followed by in situ hybridisation , we followed the procedure described by Manoukian and Krause (Manoukian and Krause , 1992) . Photographs were taken using Nomarski optics or a SP1 Leica confocal microscope .

786 RESEARCH ARTICLE Development 133 (5)

Preparation of embryonic cuticle

For the analysis of embryonic cuticle , late embryos were removed from the chorion and vitelline membrane , and mounted in a mixture of Hoyer&#x2019;s medium (van der Meer , 1977) and lactic acid (1:1) .

Artemia , Parhyale **and crayfish immunostaining and in situ hybridisation**

We initially cloned a fragment of *vvl* from *Artemia* , by PCR from cDNA generated from larval RNA with degenerated oligonucleotides designed from the *Drosophila vvl* sequence . An antibody against the Vvl protein of *Artemia* was generated by injecting rabbits with a His-tagged fragment of the Vvl protein [amino acids 241 to 386 of the previously published sequence (Chavez et al. , 1999)] . The serum was then affinity purified on a nickel column . We initially cloned a fragment of *vvl* and *trh* from *Parhyale hawaiensis* by PCR from a cDNA library kindly provided by Nipam Patel (University of California , Berkeley) , with degenerated oligonucleotides designed from the *Drosophila vvl* and *trh* sequences . We obtained a 240 bp fragment of the *Parhyale vvl* gene that we used to clone the full-length cDNA from the library . We also obtained *a Parhyale trh* fragment of around 700 bp that encompasses the region of the HLH , the PAS-1 and the PAS-2 domains (corresponding to amino acids 100 to 550 in *Drosophila*) . In situ hybridization in *Parhyale* was carried out using a protocol provided by Nipam Patel ; the protocol is available upon request . For immunostaining in crayfish embryos , we used the 4D9 monoclonal antibody for Engrailed (

-Gal4	please verify this token
-Gal4	please verify this token
DN	please verify this token
Pai	please verify this token
Developmental	please verify this token
Cappel	please verify this token
Tautz	please verify this token
anti-sense	please verify this token
Manoukian	please verify this token
F	please verify this token
2B,D,G	please verify this token
Keilin&#x2019;s	please verify this token
3K	please verify this token
&#x2013;	please verify this token
3M	please verify this token
Brusca	please verify this token
-expressing	please verify this token
Mitchell	please verify this token
Crews	please verify this token
4C-E	please verify this token
Lankester	please verify this token
Vvl	please verify this token
F	please verify this token
appendages/legs	please verify this token
vvl/trh	please verify this token
green	please verify this token
endopod/leg	please verify this token
Martinez-Arias	please verify this token
even	please verify this token
Tijana	please verify this token

Fig. 3.4 FlyBase data curating GUI.

3.6 Summary

This chapter presented the need for domain-specific NERs by presenting two case studies on how domain-specific NERs are essential. The cases presented are in two different domains. The first case study is in the digital humanities, mainly for archaeological text. The second case is in the biomedical domain in a very specialised domain, detecting NEs for Drosophila genes.

The domain-specific NERs discussed above used AL as method to improve performance and reduce expenses. Using AL for general-purpose NERs and for domain-specific NERs has proven to be a useful method. Therefore, reviewing recent work on AL is important as part of the literature review of this thesis. A detailed review of the field of AL is provided in the following chapter (Chapter 5).

Chapter 4

Crowdsourcing

Crowdsourcing, as defined by [Howe, 2006], is the process through which a group of people in an open cell solve a problem. The Cambridge Dictionary defines crowdsourcing as a verb, meaning “to give a task to a large number of people or to the general public, for example, by asking for help on the internet, rather than having tasks done within a company by employees”. Today, crowdsourcing is implemented as a business model that supports collective and distributed problem solving. People from different places worldwide can contribute to tasks and collectively finish them. Tasks are broken into small subsets that, when combined, have a huge impact on the focal problem. This process significantly reduces cost, time and effort. One very popular example of crowdsourcing is Wikipedia[1].

Companies around the world seek to introduce solutions to automate trivial tasks that are not difficult, but take significant time to accomplish. Image recognition, for example, is a problem that has lacked a proper solution for decades. Similarly, optical character recognition (OCR) [Mori et al., 1992] has posed such difficult challenges that many researchers in this field have given up. Though such issues seem trivial for humans to tackle, they have proved extremely difficult for computers.

[1] https://wikipedia.com

The internet has significantly impacted the development of crowdsourcing, supporting the use of several approaches to solve problems. Crowdsourcing participants can provide content, tag pictures and places, engage in games with purpose, and much more. In addition, several platforms built from scratch (e.g. Amazon's Mechanical Turk [mTurk2[2]] and Crowdflower[3]) have enabled people to crowdsource others' tasks. As more such platforms have become available, it has become very easy not only to crowdsource tasks, but also to conduct research in this field.

Crowdsourcing is an interesting research area that has attracted all types of research. Many researchers have focused on studying the impact of crowdsourcing and analysing how it performs [Howe, 2006] [Kittur et al., 2008]. In particular, many scientists have examined how workers behave in crowdsourced platforms, studying, for instance, what motivates people to take part in crowdsourcing platforms or what levels of quality can be gained from such platforms [Aker et al., 2012]. The following sections discuss these topics.

4.1 User motivation

Websites' user interfaces play a big role in user experience. This is also the case with crowdsourcing, since workers' completion of others' tasks can be affected by the design of the interface [Ipeirotis, 2010a]. This, among other factors, has contributed to the success of many crowdsourcing platforms. Wikipedia, for instance, facilitates information sharing and editing and makes it very easy for anyone in the world to participate. Specifically, it makes participating easy and lowers barriers to entry, allowing people to contribute on whatever level they like. In general, in crowdsourcing, when tasks are smaller, people are more likely to participate [Grady and Lease, 2010].

[2] https://www.mturk.com

[3] http://www.crowdflower.com

The motivation for crowdsourcing participants varies; therefore, studying such motivation is a difficult task. It has been found that many crowdsourcing participants stop engaging after only a few submissions [Yang et al., 2008]. Moreover, participants' demography—and, thus, their motivations—can change over time. For example, though contributors to Amazon's Mechanical Turk (mTurk) project used to work primarily in the U.S., they are now based largely in India [Ross et al., 2010]. This shift also indicates a transition from moderate-income workers to international and well-educated workers. Moreover, the majority of today's mTurk workers are younger than they were before [Ross et al., 2010].

Changes in the demographics of crowdsourcing platforms, such as mTurk, also affect the quality of the work. Many modern workers rely on such platforms for income [Ross et al., 2010], leading them to focus on quantity rather than quality. As a result, the quality of work on such platforms is lower than the quality of work done by professionals or even students [Gillick and Liu, 2010].

However, there are always people willing to take part in crowdsourcing, especially when there is something to motivate them, such as competing against others [Yang et al., 2008]. Crowdsourcing participants may be motivated to participate by higher rewards or a perceived greater chance of "winning" (e.g. due to fewer opponents) [Downs et al., 2010].

Among other reasons, people take part in crowdsourcing platforms for two main reasons:

4.1.1 Extrinsic reasons

This is perhaps the primary reason people take part in crowdsourcing platforms. Crowdsourcing has become a primary income source for many people, especially people in India [Silberman et al., 2010]. Studies show that more than a quarter of workers rely solely on crowdsourcing platforms as their main source of income [Ipeirotis, 2010b], compared to just over 10% of workers in the U.S. This financial dependence has attracted significant research. However, engaging in crowdsourcing for financial reasons is not always a negative thing. With a properly designed task and well-targeted workers, increasing financial rewards can actually increase the quality of the work [Harris, 2011]. Conversely, very low payments can attract low-quality work and spam [Kazai, 2010].

Therefore, there should always be a balance between crowdsourcing payments and tasks [Horton and Chilton, 2010] [Moreno et al., 2009]. Such a balance can increase the amount of crowdsourcing work done and improve the overall quality of the work.

4.1.2 Intrinsic reasons

Money is not the only motivation for people to participate in crowdsourcing. In fact, the first participants in crowdsourcing were not motivated by money [Zheng et al., 2011]. Platforms like Wikipedia and other successful websites functioned for years with no financial rewards for participants. However, reward systems may encourage greater participation [Chamberlain et al., 2008]. For example, the Stack Exchange network of platforms offers rewards in the form of what it calls "reputation" [Movshovitz-Attias et al., 2013]. Every time someone has his or her participation recognised by others, he or she receives points. The points depend on such measures as the number of times a question is viewed or the number of up-votes it receives. Participants also receive points for answering questions.

The reward model has been adopted and shown to be successful in many other crowdsourcing platforms. The more attention a participant gets, the more he or she is willing to stay engaged [Huberman et al., 2009]. YouTube[4] is another example in which participants—in this case, people who post videos—can track levels of engagement

[4] https://youtube.com

(e.g. by monitoring the number of views of their videos and the number of up- or down-votes each video receives). Participants who receive more attention are more likely to participate, while participants who receive less attention are less likely to participate [Wu et al., 2009]. In addition to engagement and reward practices (e.g. reputation), there are several other factors that drive participation. Passion and a desire to learn new things drive a number of people to take part and participate [Nam et al., 2009]. In fact, self-motivated participants can produce even higher quality work. However, for work to be considered high quality, competition with others is necessary. Such competition can grow harder over time [Archak, 2010]; however, it can also produce higher quality work.

4.2 Application of crowdsourcing

The two main types of participants in any crowdsourcing platform are requesters and workers. The requester submits a task, and the worker does that task. Although this might seem simple, significant effort is required to analyse what sorts of task can be submitted and completed with a reasonable amount of effort and time. Tasks can be broken down into mini-tasks and these mini-tasks can be split even further. The reason for this is that workers are paid per task [Karger et al., 2014]. The smaller a task is, the smaller the payment can be. Workers are also more likely to engage with smaller tasks.

Current crowdsourcing platforms try to use social aspects to engage workers and requesters. For example, workers are often rated on performance [Welinder and Perona, 2010]. Requesters are also rated on, for instance, their commitment to payments. Both requesters and workers can also review each other. This helps both sides, since workers with higher ratings can ask for more money to complete tasks. Requesters can also seek higher-rated workers to ensure quality. Ratings and reviews give requesters and workers credibility, showing that both are serious about the crowdsourcing environment.

Several studies have been undertaken to statistically predict the time required to complete a crowdsourced task [Voyer et al., 2010]. This particular study shows how each task parameter (e.g., type of task or reward given) can affect time. One interesting area of research, especially in gaming platforms, involves workers' behaviours [Singh et al., 2009] and how they take part in tasks. For example, research has been done on multiple game-centric human modelling [Jain and Parkes, 2009].

This extends the competitive nature of such platforms. Studies show that participation rates can improve logarithmically as a function of the reward offered [DiPalantino and Vojnovic, 2009]. Other studies have used algorithms to try to predict the quality of each worker [Ipeirotis et al., 2010]. Additional efforts have sought to address this issue by calculating errors for relevant judgements [Carterette and Soboroff, 2010] and exploring how this can affect workers' quality.

Although the general concept of crowdsourcing is fairly static, its applications in different fields differ, as do the ways in which different people contribute to different types of tasks. The following section will address some of the areas in which crowdsourcing has been implemented. The different applications for crowdsourcing are as follows:

4.2.1 Subjective tasks

Subjective tasks include reasoning and opinion-mining tasks, in which workers are asked to solve problems that are difficult for computers to solve [Jagadeesan et al., 2009] [Heer and Bostock, 2010]. For example, interpretation has always been a task that only humans are good at (though, if computers were able to exploit humans' ability, they may be able to do interpretation tasks in the future). Gathering opinions in the real world can be difficult; however, crowdsourcing can help [Mellebeek et al., 2010]. Algorithms can be used to normalise gathered information, since opinions are subjective and not all workers have the same opinions about a particular task. In fact, studies on opinion mining, particularly with respect to common sense knowledge [Gordon et al., 2010] [Wang and Callison-Burch, 2010], show that people have different views on crowdsourcing. This is the case with relevance evaluations, one of the first tasks for which crowdsourcing was used [Alonso et al., 2008]. In relevance evaluations, workers examine a number of documents, images, audio clips or other media to determine their relevance to a specific query or list of queries. One platform for accomplishing this is CrowdSearch[5] [Yan et al., 2010a], which uses crowdsourcing to power a search system, using the crowd to validate the relevance of the search results. Subjective tasks also include natural language tasks, such as word sense annotation [Akkaya et al., 2010], and more basic tasks, such as spam filtering[6].

4.2.2 Objective tasks

Objective tasks require workers to select one or more items from a list of options. They may either select one or more correct answers or rank a list of items. Objective tasks are perhaps the most popular types of tasks because they are simple and produce a clear answer. However, the usefulness of a task depends on how the task is designed and how precisely it is implemented. If the options are ambiguous, then it is impossible to guarantee high-quality answers. Natural language processing has benefited significantly from crowdsourcing [Finin et al., 2010] [Demartini et al., 2012], drawing responses from non-expert resources that would otherwise be very expensive to acquire. Similarly, crowdsourcing has been shown to outperform expert work on word alignment [Gao and Vogel, 2010].

[5] http://crowdsearch.me

[6] http://sourceforge.net/projects/razor

4.2.3 Data gathering

All of the types of tasks mentioned above can be considered data gathering tasks; however, here, we refer to the generation of data subjective to the worker. The tasks in this section are normally designed to gather new data, rather than to validate, process or correct existing data. They represent perhaps the very first type of crowdsourcing. Crowdsourcing first became popular through Web 2.0 technologies [Yuen et al., 2011] [Yuen et al., 2009]. Successful stories include Wikipedia[7], an online encyclopaedia whose extensive and varied content is provided by members of the general public from all over the world. The website del.icio.us[8] is also considered one of the most successful crowdsourcing platforms. On this website, users link to all sorts of information and give tags for every link shared. There are also several question-and-answer platforms, including both general-purpose platforms (e.g. Yahoo Answers[9] and Answers[10]) and more domain-specific platforms (e.g. the large number of platforms belonging to Stack Exchange[11], such as Stack Overflow[12] for programmers). Stack Exchange has demonstrated the successful implementation of such platforms, and its network includes more than 20 specialised portals for such topics as mathematics, physics and linguistics.

Crowdsourcing has also been used to generate corpuses (e.g. for speech and language applications [Callison-Burch and Dredze, 2010]) and create resources, such as a phrase attachment corpuses [Jha et al., 2010], translations and dictionaries [Parent and Eskenazi, 2010]. One task that attracted significant media attention was The Sheep Market[13] [Koblin, 2009], which asked people to draw a sheep facing left. There

[7] https://wikipedia.com

[8] https://del.icio.us

[9] https://yahoo.com/answers

[10] http://answers.com

[11] http://stackexchange.com

[12] http://stackoverflow.com

[13] http://comparethemarket.com

have also been cases of successful businesses based on crowdsourcing. For example, *Threadless*[14] crowdsources t-shirt designs from creative designers.

4.3 Gamification

Gamification is the use of game theory and design in tasks that are not normally considered games. Players are rewarded for doing tasks by, for instance, levelling up, securing a spot on a leader board or receiving stars. Gamification methods seek to encourage more people to engage and to motivate current players to continue to engage and to engage at higher levels. The main difference between gamification and “real” games is that in real games, you can win or lose, whereas in gamified situations, you always achieve some sort of reward (e.g. a new level or points). This does not mean that gamification is fundamentally different from traditional games; in fact, most games actually begin with gamification (e.g. to get players engaged) and then move on to more game scenarios. The term “gamification” has been used for decades to mean different things. In the past, it meant “turning something into a game”; however, it now means the process of “turning a game into something not a game” [Bartle, 2016]. This fundamental difference can make it difficult for non-game designers to adapt gamification. For example, many design games simply give users points for doing tasks. This approach can be called Pointsification [Robertson, 2010].

As with crowdsourcing, people take part in gamified platforms for both intrinsic and extrinsic reasons. To some extent, playing the game itself can be considered a source of intrinsic motivation, since people play games for fun and entertainment. Other users may focus on the extrinsic rewards gained by completing tasks. To engage users, gamification often uses extrinsic rewards.

[14] https://www.threadless.com

4.4 Quality control

Controlling the quality of crowdsourcing is a challenging task. Crowdsourcing environments have a social nature, and people expect to have no boundaries. This is not always negative, since it suggests that there is no limit to creativity. However, when a crowdsourcing community (e.g. mTurk) is built artificially, the absence of regulations makes quality assurance challenging. Many studies have examined crowdsourcing environments. Crowdsourcing quality is affected by many aspects, such as money, design, and task length. However, each of these conditions has particular constraints and intersects with other conditions. For instance, increasing payments for well-designed tasks can increase quality [Harris, 2011]; however, if quantity is more important, increasing payments may increase quantity, but not quality [Mason and Watts, 2010].

Due to the importance of quality control, many have tried to automate and integrate quality assurance into the very design of tasks posted on such platforms as mTurk [Lease, 2011] [Tang and Lease, 2011]. The behaviour of crowdsourcing workers, although difficult to predict, can still predict quality. Some studies have explored this by training a model on crowdsourcing task output [Huang et al., 2010]. The model then predicts not only the quality, but also the design of the tasks. Though this works in some cases, it may not be feasible for more complex tasks. Therefore, other researchers have introduced hybrid integration of experts and crowdsourcing. One particular study [Voyer et al., 2010] used this approach to examine named entities. First, before the crowdsourcing started, an expert labelled all of the data. Using a hybrid approach increases the quality of entities found in the data, such that crowdsourced workers need only to classify these entities into types.

Quality control investigations have two primary objectives: to maximise quality and minimise errors. Some studies have focused on the minimising errors part of this equation, resulting in the transfer of several approaches from traditional research methodologies on cheat detection (e.g. the use of control questions or even control groups [Eickhoff and de Vries, 2011]) to the field of quality control in crowdsourcing. These methods focus on using crowdsourcing to achieve financial gain rather than to produce quality work. However, given the huge number of participants in crowdsourcing, more automated techniques are required. Since the cost of using crowdsourcing is low, crowdsourcing can be used for evaluation through, for example, a majority decision or a control group [Hirth et al., 2010]. In the control group approach, the work done by one group is checked against, compared to or combined with the work of another group. In the majority decision approach, the same task is given to more than one group. Then, their answers are aggregated, and the majority makes the decision. Alternatively, an evaluation can be done by submitting a task to only one group and then asking a different group to evaluate their work.

In many cases, however, such measures do not always guarantee quality, especially in cases when the task itself is somehow ambiguous or difficult. As mentioned earlier, ensuring that a task is designed properly can ensure a higher level of quality. Task design involves several factors, including choosing the right task, designing an appropriate interface, targeting the right audience, and, lastly, implementing an evaluation measure that considers everything mentioned before.

4.5 Expert sourcing

Expert sourcing is crowdsourcing with experts. It is more commonly known as a business model than as an area of academic research [Woolley et al., 2015]. This may be because most academic research requires expert involvement, and thus cannot be considered expert sourcing. For example, in the NLP field, creating resources (e.g. a named entities data-set) requires the immediate availability of experts to carry out the annotation.

Still, the concept of expert sourcing —and, more specifically, having expert workers in crowdsourcing— has attracted some research interest. For example, in heterogeneous crowdsourcing platforms, studies suggest classifying workers based on their expertise before assigning them tasks [Raykar et al., 2010]. The idea is that, regardless of the workers' expertise, they are not guaranteed to know every aspect of the task in hand. To ensure that each worker completes the task about which he is most knowledgeable, tasks should be divided into even smaller parts.

Expert sourcing, or the consideration of crowd expertise, has also attracted some research. In fact, more than one stream of research has taken advantage of the information we know about workers' expertise. For example, active learning research [Yan et al., 2011] has incorporated crowdsourcing in many ways, specifically for NLP tasks, though there is still a lack of research combining expert sourcing and active learning. Combinations of crowdsourcing and active learning have been used in sentiment analysis [Brew et al., 2010], object recognition in images [Vijayanarasimhan and Grauman, 2014], classification [Costa et al., 2011], machine translation [Ambati et al., 2010] [Ambati, 2012] and many more. As mentioned earlier, although expert sourcing has gained some recent attention, due to the complexity of expert sourcing, it has not been extensively studied.

The popularity of crowdsourcing can be seen from the discussions in the previous sections. Crowdsourcing is a resource-saving method for creating and extracting knowledge, and it has been applied to many research areas, especially related to natural language processing. Studies have explored topics ranging from how crowdsourcing works to why people take part in such environments. The main benefit of crowdsourcing is that it reduces the cost and effort required to create resources. This thesis will build on the findings of other studies on crowdsourcing. The following section presents a different method for reducing the effort required to create resources and extract knowledge: namely, active learning. This method has been widely used to annotate resources, especially for under-resourced domains, such as the digital humanities [Ekbal et al., 2011].

Chapter 5

Active Learning

Active learning (3.3) is a method to reduce the effort required in annotation by designing a loop that begins with a small data-set and then continues with the selection of the most informative examples. With enough annotated examples, a model can be retrained to refine its sample selection decision. Compared to random sampling, an approach involving less annotation can achieve greater performance [Laws et al., 2011]. Active learning has been used in many natural learning tasks, such as:

- Co-reference resolution [Sachan et al., 2015] [Laws et al., 2012] [Gasperin, 2009] [Miller et al., 2012] [Zhao and Ng, 2014].

- Information extraction [Finn and Kushmerick, 2003] [Cardellino et al., 2015a] [Kholghi et al., 2015] [Hady et al., 2014] [Cardellino et al., 2015b].

- Machine translation [Logacheva and Specia, 2014] [Du et al., 2014] [Bloodgood and Callison-Burch, 2010] [Ambati et al., 2010] [Gangadharaiah et al., 2009] [Haffari et al., 2009].

- Morphological segmentation [Grönroos et al., 2015].

- Named entity recognition [Chen et al., 2015] [Kim et al., 2006] [Ekbal et al., 2012b].
- Part-of-speech tagging [Nepil et al., 2001] [Ringger et al., 2007a] [Ringger et al., 2007b].
- Relation extraction [Zhang et al., 2012].
- Sentiment analysis [Smailović et al., 2014] [Hajmohammadi et al., 2015] [Kranjc et al., 2015] [Li et al., 2012] [Li et al., 2013b] [Brew et al., 2010] [Smatana et al., 2013].
- Speech recognition [Yu et al., 2010] [Hakkani-Tur et al., 2002] [Zhao and Ma, 2013] [Riccardi and Hakkani-Tür, 2005].
- Speech summarisation [Zhang et al., 2009] [Zhang and Fung, 2012].
- Word sense disambiguation [Alagić and Šnajder, 2015] [Zhu et al., 2008] [Zhu and Hovy, 2007] [Chen et al., 2006] [Chan and Ng, 2007].

The key idea of active learning is that if we allow a machine learning algorithm to choose what to learn from, it can perform better [Settles, 2012]. In the process of active learning, an algorithm has a specific query method that it uses to select samples to be labelled by an annotator. This is an iterative process, such that both the algorithm and its query method improve over time. In other words, with more iterations, the algorithm can refine its query method to select even better examples.

An active learning process involves the formation of several elements. These elements include scenarios involving several different components, which serve as general frameworks for the active learning process. Active learning also involves a query strategy, which is the heart of the active learning process. This query strategy defines how samples are selected for annotation. Finally, active learning requires a human agent to carry out the annotation. The active learning process stops, according to a stopping criterion. The following sections will address the various elements that support active learning. Several scenarios are presented, and a number of well-established selection criteria are discussed. Finally, we conclude with a section on stopping criteria.

5.1 Scenarios for active learning

In active learning, the ways in which queries are asked, examples are provided and the annotator provides feedback are collectively called the scenario. This means that the ways in which the data are given to the algorithm and the query method can differ. This section will detail the most common scenarios of active learning.

5.1.1 Online sampling

Online sampling is sometimes called stream-based or even selective sampling [Moskovitch et al., 2007] [Thompson et al., 1999]. This type of sampling was one of the first active learning scenarios studied [Cohn et al., 1994] [Cohn et al., 1994]. It assumes that attaining an unlabelled instance is inexpensive or free; hence, sampling can be done from an actual distribution, in which the learner decides whether or not to label the instance. In such online active learning, the samples come from unlabelled instances provided by an input source at different intervals. The learner then decides whether to discard or query each instance. In this scenario, active learning assumes that the input source comes from an unknown and non-uniform distribution; thus, the query strategy is continuously refined, based on the underlying distribution of the input data.

The decision to discard or query each instance is established through diverse sampling strategies. For example, one such strategy involves defining an information content or utility measure while making a random biased decision to query instances with higher utility [Dagan and Engelson, 1995]. Another mechanism involves the computation of an explicit region of uncertainty, which comprises a component of the instance-ambiguous space for the learner, while focusing on the instances available within it for querying. Similarly, a minimum threshold may be set on the measure of utility defining the region. This allows for the querying of the instances with evaluations above the threshold [Pedrycz et al., 2010].

5.1.2 Pool-based sampling

In this scenario, samples are selected from a pool that is assumed to be closed, non-changing, and static. The choice of queries follows a strict design, based on the utility measure, which is based on the evaluation of all pool instances, including [Balasubramanian et al., 2014].

The pool-based scenario has been studied in diverse machine learning contexts, such as information extraction, text classification, image classification or retrieval [Qi and Zhang, 2016] [Zhang et al., 2014], video classification or retrieval, speech recognition and the diagnosis of cancer [Reker and Schneider, 2015]. It is considered the most preferred scenario for applied research active learning. This may be because it facilitates the gathering of large collections of unlabelled data within the domain of real-world learning problems [Lewis and Catlett, 1994], thanks to the internet.

5.2 Query strategies

Both online active learning and pool-based active learning rely on a sampling method that follows a utility function to determine what instance to select for annotation. The sampling method can vary according to different approaches. The following sections discuss a number of query strategies.

5.2.1 Uncertainty sampling

Uncertainty sampling allows an active learner to query the instances with least certainty among all other instances [Lewis and Gale, 1994]. Uncertainty sampling is a very popular query strategy that has been used in many areas of NLP, including, especially, statistical sequence modelling (e.g. named entity recognition [Lewis and Gale, 1994] [Lewis and Catlett, 1994] and binary classification [Settles, 2012]).

Uncertainty sampling measures the uncertainty of a model at a given point. This can be done in many ways, including *least confidence* sampling [Lewis and Gale, 1994], *margin* sampling [Kim et al., 2015], and *entropy-based* sampling [Shannon and Weaver, 2015]. For example, in binary classification problems, a probabilistic model can use uncertainty sampling to query instances in which the posterior positive probability is near 0.5. Multi-class classification implies the same idea: sampling the instances about which the current model is the most confused [Settles, 2012].

Each of the mentioned uncertainty sampling strategies has pros and cons. For example, the *least-confidence* strategy may only consider the information on most probable labels. Therefore, it effectively loses the information regarding the distribution of remaining instances. This can be overcome by the use of *margin sampling* [Kim et al., 2015], which aims to correct the limitations of the *least-confidence* strategy by incorporating the second most likely label. Ideally, the instances with large margins are considered easy because the classifier properly differentiates between the two most likely class labels. In fact, the instances with small margins tend to be more ambiguous, leading to a need for full knowledge of the true label, and thus aiding the model's effectiveness in discriminating between the two options.

On the other hand, many tasks, such as named entity recognition, tend to have large label sets, meaning that the margin approach will ignore much of the output distribution for the remaining classes. Thus, *entropy* is used as an uncertainty measure and a more general uncertainty sampling strategy [Shannon and Weaver, 2015]. The entropy-based approach aids by generalising complex structured instances, such as sequences or trees, to probabilistic models and probabilistic multi-label classifiers.

The strategies for uncertainty sampling can also be applied using non-probabilistic classifiers. The decision tree classifier was the first one used to explore uncertainty [Lewis and Catlett, 1994]. Even simple nearest-neighbour classifiers can been used in active learning: For example, every neighbour can be allowed to vote on the class label, and the proportion of the votes can represent the probability of the posterior label [Yang et al., 2015]. Uncertainty sampling has also been used with support vector machines (SVMs) [Vlachos, 2004]. However, the approach is equivalent to uncertainty sampling with probabilistic binary linear classifiers (e.g. logistic regression). The concept of uncertainty can also be used in regression problems. For example, learning tasks can be used with continuous output variables instead of discrete class labels [Li et al., 2013a]. In that regard, the learner can query the unlabelled instance with the highest prediction output variance. Furthermore, it can be used for Gaussian random fields [Ma et al., 2013]. Here, the assumption is that the entropy of a random variable may be its variance's monotonic function; hence, the approach is similar to the entropy-based uncertainty sampling approach utilised in the classification.

5.2.2 Query-by-committee

The query-by-committee (QBC) algorithm is a query selection framework motivated by the theory of allowing committee members to vote on query candidates [Seung et al., 1992]. Ideally, the most informative query is the one for which all committee members had the most disagreeing instances. The aim of the QBC approach is to query the controversial regions of input space.

Query-by-bagging has been used for many non-probabilistic problems, such as ensemble learning techniques (e.g. boosting and bagging) to construct a committee [Zhang et al., 2015]. The literature regarding appropriate committee sizes has failed to reach consensus; thus, applications of this approach vary [Fu et al., 2013]. However, the literature has not overlooked small committee sizes [You et al., 2014]. For example, the pool-based margin strategy can be used with SVMs to minimise version spaces by weighting instances [Bouguelia et al., 2016]. Alternatively, a selective sampling algorithm may utilise a committee of two distinct neural networks: the most general and the most specific models at the two extremes of the version space [Cohn, 1994] [Hanneke, 2014]. The QBC approach can also be applied to regression settings; for example, it can be used to measure disagreement regarding the variance among committee members' output predictions [Di and Crawford, 2012].

5.2.3 Expected model change

The expected model change sampling strategy selects instances with the greatest impact on the current model [Settles and Craven, 2008]. The strategy was developed for active learning in multiple-instance situations. The perception behind this framework is based on preferred instances that can influence the model, despite the resulting query label. Ideally, this approach works well in empirical studies, even though it may be computationally expensive, especially in situations of large labelling and feature space. However, this approach also struggles with poorly-scaled features that hamper information levels through overestimation due to unusually large feature values or large corresponding parameter estimates [Settles, 2012].

This strategy has been implemented in several studies, including studies of probabilistic sequence models with CRFs [Vezhnevets et al., 2012], gradient-based optimisation

[LeCun et al., 1998] and parameter value vector re-estimation [Huang and LeCun, 2006].

5.2.4 Expected error reduction

The expected error reduction approach is somewhat similar to the expected model change approach; however, it targets and measures the level of the reduction of the generalisation error rather than the level of change in the model [Roy and McCallum, 2001]. Theoretically, this approach can be used to optimise generic performance measures, including maximisation of the area under the ROC curve, F1 measures, recall, precision, and loss value minimisation. It is a computationally expensive query framework, since it requires the estimation of the expected future error for every query and the incremental retraining of the model for every possible query labelling.

The error reduction framework was first proposed for text classification by Naive Bayes. It was then combined with the semi-supervised learning approach to produce improvements in uncertainty sampling [Zhu et al., 2003]. Uses also include logistic regression [Li and Guo, 2013], Gaussian random fields [Zhu et al., 2003], support vector machines [Cai et al., 2014] and non-parametric models, including the Gaussian random fields [Zhu et al., 2003].

5.2.5 Sampling strategy shortcomings

A critical aspect of the variance reduction frameworks and the estimated errors is their focus on input spaces instead of individual instances. They have less vulnerability to query outliers than other query strategies, such as EGL, QBC, and uncertainty sampling. The EGL and QBC approaches are similar in terms of behaviour, especially with respect to the querying of possible outliers based on their expected controversy or impact on the model. The variance reduction and estimated error strategies avoid problems by using the unlabelled pool to estimate output variances and future errors. Problems can also be overcome by explicitly modelling the input distribution when selecting a query [Cai et al., 2015].

5.2.6 Stopping criteria

Knowing the appropriate time to stop learning or posing queries is a potentially important attribute of active learning applications [Bloodgood and Grothendieck, 2015]. Stopping criteria focus on the best mechanism for thinking about an issue from a critical perspective, in which the cost of acquiring new training data supersedes the cost of errors made by the current system. This approach studies the points at which a learner's accuracy reaches a plateau, and the acquisition of more data may waste resources. It involves techniques by which the active learner can stop asking questions in order to save resources.

In general, stopping criteria stop the loop of active learning by when the performance of the classifier, as measured using an available test, fails to improve (at least to a satisfactory rate). Measuring performance can involve measuring the quality of the classifier (e.g. the F-score) or measuring its confidence [Vlachos, 2008].

Chapter 6

Thesis Contribution

The previous chapters have given the reader an overview of the multidisciplinary nature of the work presented in this thesis. The first chapter introduced the field of NER. It took a different approach to most literature surveys, as it focused on the history of NER. Most NER literature reviews list the current or the most recent literature in the field. This approach would normally answer questions like *What?* and *How?*. It would explain, for example, the recent algorithms for NER, or how an algorithm has been applied. Although the approach of most NER literature reviews is useful to give an overview of the state-of-the-art work in NER, as is the case in many literature reviews for other knowledge domains, it would normally fall short of answering the question *Why?*.

Therefore, the reason for focusing on the history of NER in Chapter 2 is that this chapter tries to answer the question *Why?*. The most important question that it answers is why newswire became the dominant domain for NER.

The discussion then narrows down from general purpose NER (Chapter 2) to domain-specific NER in Chapter 3. Narrowing the scope is to show how each knowledge domain has its own characteristics when it comes to NER.

Chapter 3 illustrates in two case studies how important domain-specific NER is. However, the two case studies presented in Chapter 3 share much in common, even though they are in two completely separate domains. They share challenges such as minimising effort and resources to develop NER. These two case studies and many others in the literature have tried to tackle the issue of having limited resources when developing a knowledge extraction system. For example, developing a supervised machine learning NER requires training data. This training data needs to be annotated by domain experts.

The use of active learning to develop NER has been extensively covered in the literature, either in the digital humanities, like the archaeology domain, or in the biomedical field, such as in gene drugs and disease detection. However, the work presented in this thesis introduces a number of novelties, which are detailed in the following sections. For simplicity and to distinguish the proposed approach introduced from others, in this thesis, the proposed approach is therefore given the name *active expert sourcing*. The name is a blend of active learning, expert annotation, expert sourcing and crowdsourcing.

The next sections will outline the research objectives and the contribution of this thesis. Much more detail is provided, compared to what has already been mentioned briefly in Chapter 1. Also extensive detail is provided for the research methodology and experimentation set up.

6.1 Research hypothesis

The hypothesis of this thesis is: **Crowdsourcing and active learning can help experts extract knowledge from domain-specific information**.

6.2 Research objectives

A number of objectives have been set for this thesis. The aim is to stimulate state-of-the-art methodologies and techniques in the field of domain-specific NER.

The research objectives are listed below. The list of objectives has already appeared in the introduction chapter (Chapter 1). However, in the following, more details are provided. Also the novelty of the contributions are justified, and compared to related work. Nevertheless, the key differences to other related work are explained. The objectives set for this thesis are:

1. To create an online end-to-end expert-in-the-loop framework for annotating domain-specific named entities.

It has been discussed in the literature review chapters how data curating tools have been adapted. However, these tools are used only offline and not in production – that is in a very controlled experimental set up. In theory, each platform might impose different challenges, and users might behave differently when using different platforms. Therefore, creating a new environment for annotating data is worth investigating.

Hypothesis 0.1

Creating an online end-to-end expert-in-the-loop framework for annotating domain-specific named entities is feasible.

Related work

This paper differs from the related work in various axes. Unlike [Vlachos and Gasperin, 2006] and Poesio et al. [2011a], the objective is to tightly integrate domain experts at the centre of the system. Unlike Poesio et al. [2011c], the objective is to have an end-to-end system where not only knowledge extraction happens but also domain expert annotation.

2. The framework developed should produce high-quality annotated data.

The majority of studies on crowdsourcing discuss the low-quality nature of produced data, which has resulted in many studies devoted to controlling the quality of crowdsourcing.

This thesis argues that when approaching the right platform for a particular domain, a better result can be gained. There are many online platforms that can be approached as a crowdsourcing platform.

Hypothesis 0.2

Approaching different platforms to current crowdsourcing platforms will yield high-quality annotated data for domain-specific NER.

Related work

Instead of trying to increase quality though payment [Harris, 2011] or use resources on integrating quality assurance methods [Lease, 2011] [Tang and Lease, 2011], this thesis takes a different approach, whereby it change the choice of platform. So instead of using current crowdsourcing platforms, the objective is to find an alternative platform where quality is “guaranteed” to be high.

3. The framework developed should be financially viable.

Developing domain-specific NER is expensive. Therefore, this thesis aims to investigate the feasibility of pushing the boundaries of research by using already available domain knowledge (e.g., methods, algorithms, and tools).

Hypothesis 0.3

Existing tools, software, methods, concepts, techniques or algorithms can push the boundaries for domain-specific NER.

Related work

Unlike Poesio et al. [2011a] and Gramates et al. [2016], where funding is/was available for developing domain-specific NER, the objective is to develop the theory for a financially viable framework.

4. The framework developed should help the development of domain-specific NER.

As creating an annotated resource is important, developing a real-world domain-specific NER is also important. Hence, this thesis aims to study the impact of deploying a real-world platform that solves real-world problems.

Hypothesis 0.4

With limited resources, a real-world production-ready platform can be deployed for domain-specific NER.

Related work

Not only the data that this thesis is gathering, as in Vlachos and Gasperin [2006], but the set of tools that is developed – is somehow similar to Poesio et al. [2011c]. However, the objective here is to have a framework developed with all the constraints/features of all the other objectives being discussed.

5. The framework developed should make it easier to iterate the development of NER when there is a need for adjustment.

Going though a number of iterations until finding an optimal method in developing domain-specific NER is commonly seen in domain-specific NER. The issue here is that this does not only require investigating NER algorithms, but involves having constant access to domain experts.

In Chapter 3, two cases were presented which show how improving an NER can be done not by improving an algorithm, but, rather, by involving domain experts.

This thesis aims to introduce an approach that helps in this regard. This approach is to allow for access to experts whenever needed. This is achieved though integrating with paper-sharing platforms. The intuition is that since the majority of domain-specific NER data-sets are constructed from published papers, it would make more sense to stay close to where these data can be found. So whenever there is a need to iterate the development of an NER, say, if there is an NE schema or requirements change, experts can be accessed easily.

Hypothesis 0.5

Paper-sharing platforms will help in maintaining sustainable access to experts to perform data annotation for NER.

Related work

As far as could be ascertained, a paper-sharing platform has never been used as an expert sourcing platform for NER.

6. To investigate reducing the burden on domain experts by annotating data through the use of crowdsourcing-like concepts.

Crowdsourcing has achieved its success by having access to large numbers of workers. As has been discussed in Chapter 4, users of crowdsourcing platforms take part for different reasons. Although extrinsic reasons are some of the reasons, it is not the only set of reasons. There are also a lot of intrinsic reasons.

Domain-specific NER requires domain experts, which not only puts pressure on NLP scholars, but also on domain experts.

This thesis aims to use some concepts from crowdsourcing and apply them to expert sourcing. It aims to reduce the effort needed by experts and make it more intuitive for them to take part in developing domain-specific NER. This is achieved by asking experts to annotate only their own work.

Hypothesis 0.6

Experts will participate in a crowdsourcing-like platform when asked to annotate their own text.

Related work

Well-designed tasks [Harris, 2011] [Voyer et al., 2010], the sense of achievement [Movshovitz-Attias et al., 2013], passion and a desire [Nam et al., 2009] are among the concepts to be utilised to encourage experts to annotate their own text.

7. To investigate reducing the burden on domain experts by annotating data through the use of active learning.

Related to the previous point, active learning (AL) is also utilised to further reduce the effort needed to develop domain-specific NER.

Hypothesis 0.7

AL will help reduce experts' efforts in annotating data.

Related work

AL has been extensively studied in literacy, and in the literature chapters, Chapter 5 has been devoted to it. The objective here is to study AL under the constraints/features of all the other objectives being discussed.

6.3 Active expert sourcing

Active expert sourcing is a method for reducing the effort to create data-sets and extracting knowledge –especially for creating training data-sets for machine leaning algorithms such as NER. It leverages aspects of crowdsourcing to get workers (e.g., experts) involved, and reduces the cost and effort for creating resources with the use of AL. It takes advantage of the widely available platforms for sharing publications among scholars, such as ResearchGate and Academia, whereby experts who take part in such platforms are "targeted" to complete small tasks – in a crowdsourcing fashion.

Experts/authors use platforms such as ResearchGate[1] and Academia[2] to share publications. Whenever an expert uploads a paper onto any such platform, they are normally asked to provide some information about their publication. For a user to

[1] https://www.researchgate.net

[2] https://www.academia.edu

be asked to provide information about one's own publication when uploading to a paper-sharing portal is something common and that scholars are familiar with. It is common practice for conferences and journal portals as well.

In the approach proposed, this thesis attempts not to introduce anything unfamiliar to scholars/authors. It tries not to introduce any new settings for data and resource curating. It tries not to introduce complicated technical software or a set of tools. It tries to make the introduced approach as simple to implement and adapt as possible. In fact, the proposed *active expert sourcing* approach will minimise paper-sharing and resources curating work-flow to be only a matter of pressing a few buttons instead of filling out a list of text fields.

Theoretically speaking, paper-sharing platforms have been a great source of data-sets, therefore paper-sharing platforms make good candidates for crowdsourcing. Crowdsourcing has proven to be a great method for data annotation, therefore its methods and techniques make good candidates for recruiting experts in paper-sharing platforms. Nevertheless, active learning has proven to be useful in reducing resources when annotating data, and therefore it is to be utilised.

Technically speaking, the benefit of *active expert sourcing* is accomplished by a processing pipeline, along with number of machine learning classifiers. The processing pipeline takes care of converting uploaded files, which normally are in PDF format, into text format. Then this text is processed to extract information that would otherwise be required to be input manually step by step. This extracted information includes authors' names and affiliations, abstracts, figures, and, most importantly, named entities. Nevertheless, experts might sometimes be asked/required to provide feedback on pieces of information that the NER is not sure about.

To conclude this section, for *active expert sourcing* to be implemented, the following are needed:

- a portal: for authors to share their publications
- a processing pipeline: for every publication shared by an author, a pipeline needs to process the content to prepare it for the next step
- a classifier: a supervised machine learning algorithm that is to extract knowledge from the resources shared
- an AL framework/algorithm: to improve the performance of the previous step(s), and reduce the effort needed by expert/scholars/authors.

The remainder of this section details the research methodology followed to study the hypotheses presented above. Also following, is the general framework for experimental set up and evaluation.

6.4 Research methodology

This section details the methods, settings and evaluation approaches used in this thesis. The first section addresses the general method used, such as how data has been handled and how evaluation is carried out, followed by a section on the experimental parameters that have been set, and how experiments have been set up.

6.4.1 Data splitting and cross-validation

A common practice in machine learning is to split data-sets into training, developing and testing sets. Throughout our experimentation examination, different data-sets are used, and these are all publicly available. Some of the data-sets come already split into subsets (e.g., training, developing and testing sets), but some need to be processed. To increase the chance of reproducibility, data-sets that are already available to be split are kept intact. However, for those that need to be split, a stratified 10-fold

Table 6.1 Data-set splitting methods and parameter settings

	Real world	Simulation
Splitting method	Already split	10-fold cross-validation[4]
Initial training size	Training set	1% of training set
Pool size	1 document	1% of training set
Sample size	50	10% of pool size

cross-validation is utilised, where for each fold, random sampling is used with no replacement. Stratified k-fold cross-validation ensures that the mean of each fold is almost equal in all folds. This ensures that each fold can be a fair representative of all the population.

6.4.2 Parameter setting

There are some interesting parameters to be set in our experiments. The most important are *i)* initial training set, then for each iteration; *ii)* pool size and *iii)* sample size[3].

These parameters can significantly impact results, each of which is a field of research on its own. While the real-world experiment in this research is based on [Ekbal et al., 2012a], the remaining simulation experiments are inspired by [Becker, 2008].

In terms of data splitting and parameter setting, Table 6.1 details how the data were presented throughout the experiments.

6.4.3 Evaluation

In order to evaluate the performance of active expert sourcing, four variant settings were introduced. The reasons for developing such settings is to have a baseline to compare to. Since the proposed approach of this thesis consists of different components, it was thought that having multiple baselines would give a better understanding of how the proposed approach would perform. The settings introduced are:

[3] Known as batch size in active learning literacy.

- $-$A$-$E, or *Crowdsourcing*: This is random selection with non-experts. In this condition, corresponding to annotation as normally carried out, the new unlabelled instances to add to the training data are selected randomly, and feedback on these instances is provided by non-expert users.

- +A$-$E, or *Active Crowdsourcing*: This is active learning with non-expert feedback. In this condition, active learning methods are used to identify instances, and non-expert users provide feedback. These new instances are thereafter added to the training data with the feedback.

- +A+E, or *Active Expert Sourcing*: This is the condition that will hypothetically lead to the best results. In this condition, active learning methods are utilised to identify instances, and expert users (i.e. archaeologists) provide feedback.

- $-$A+E, or *Expert sourcing*: This is active learning with expert selection. This condition is meant as an upper bound. In this set up, the most problematic cases are identified by the expert(s) themselves.

Throughout the experiments, we reported the F-score [Rijsbergen, 1979], which is a harmonic mean of precision and recall.

To test the significance of the results, the data was statistically examined. It is worth noting that F-scores are usually not normally distributed, as shown in the kernel density (Figure 6.1) and histogram (Figure 6.2).

As more than two conditions were compared, the Friedman test was used to check the significance of the difference between the four settings, and the Wilcoxon signed rank test was run for post-hoc paired comparisons. Moreover, the Bonferroni correction [Weisstein, 2004] was also taken into consideration, where the critical $p - value$ (α) was divided by the number of comparisons being made. As four settings were involved in the comparison, the significance was reported only when p was $< \alpha/6$.

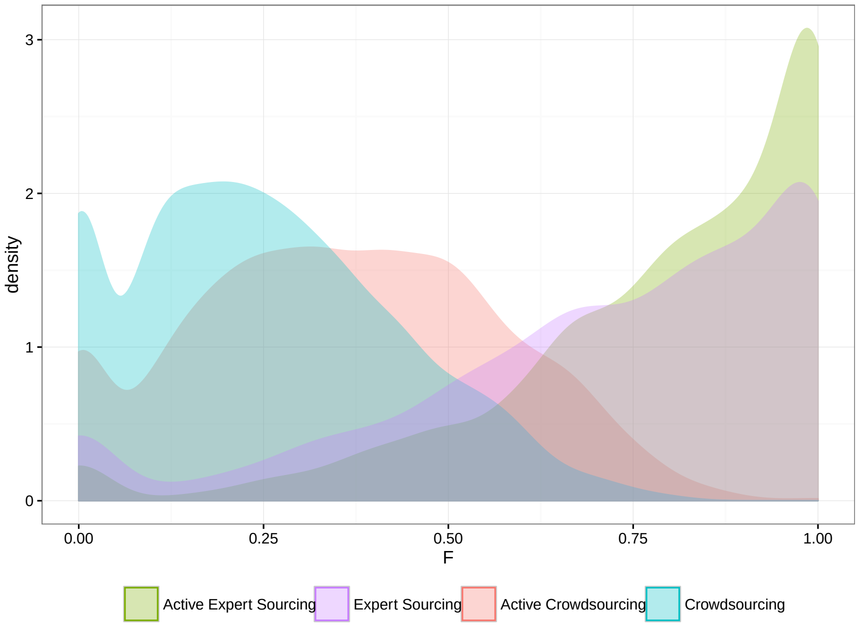


Fig. 6.1 An example of the F-score density from one of the experiments

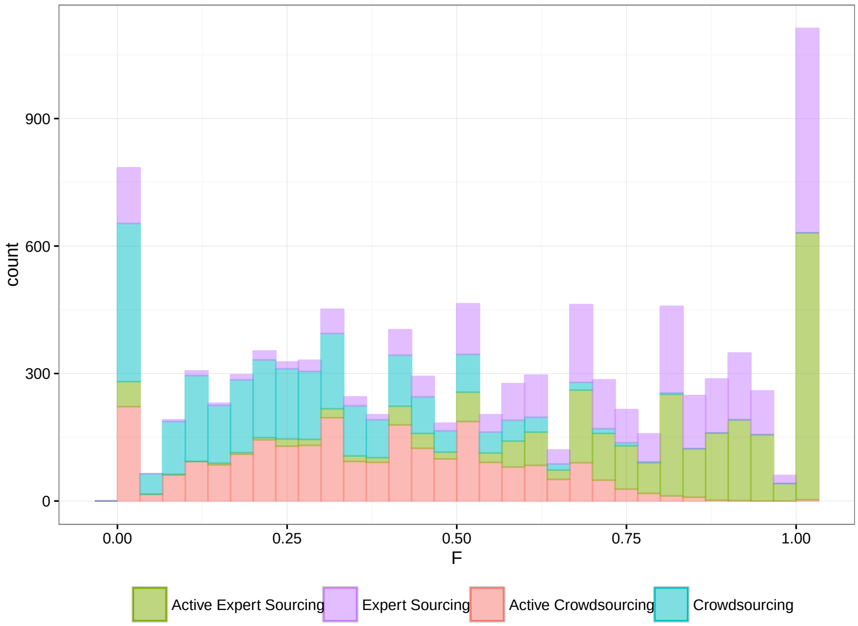


Fig. 6.2 An example of the F-score histogram from one of the experiments

6.5 A real-world experiment

In this experiment, we build on top of the work conducted by [Ekbal et al., 2012a], where a platform APSAT/ALPINET is under development as a collaboration between the University of Trento's archaeology Lab and a number of other research centres in the ALPINET network to carry out research on alpine archaeology. The portal, structured around WebGIS technology and a large database, enables scholars both to share papers and to visualise the sites of excavation discussed in these papers. Uploading these papers, however, requires scholars to enter a great deal of information (e.g., citation information, and sites and cultures mentioned). The ultimate goal of that work [Poesio et al., 2011b] is to develop methods to automatise this process of extracting information from the papers as far as possible, and also to develop novel visualisation techniques, allowing non-experts easier access to this information (e.g., spatial/entity/temporal based browsing) in addition to standard keyword search. To this end, [Poesio et al., 2011c] have been developing a pipeline able to extract much of this information automatically.

6.5.1 Quality control

In our experiment, each participant completes an entirely different task from the others. Therefore, it is almost impossible to quality control with a traditional inter-rater reliability test such as Cohen's kappa, Fleiss' kappa or Krippendorff's alpha. Therefore, the environment was controlled by allowing only privately invited participants to take part.

6.6 A simulation experiment

In the simulation experiment, the aim is to mimic the setting of the real-world experiment. We simulate a portal to which simulated participants submit data – as if it were their own publication. However, in this experiment, there are more parameters to be set, which will be discussed later in this chapter. One interesting parameter here is to set the noise level for simulated participants. It was found that this parameter does affect the results, as will be illustrated later in Chapter 8.

6.6.1 Formalisation

Having the training data $\mathcal{D}$ with N samples $\{x_1, \ldots, x_N\}$ where x_i has a set of features $[x_1^i, x_2^i, \ldots, x_n^i]$ and $x_i \in \mathbb{R}^D$, and having T annotators, where each sample is labelled by only one annotator t and is labelled only once, the label given by annotator t for the i^{th} sample is $y_i^{(t)} \in \mathcal{Y}$. The gold-standard answer for the i^{th} sample is $z^i \in \mathcal{Z}$.

Now, say for a variable, a $y^{(t)}$ that is random over the space $\mathcal{Y}$, is given by the annotator t where $t = \{1, \ldots, T\}$. As above, having randomly defined variables $x \in \mathcal{X}$ representing a known input, and $z \in \mathcal{Z}$ representing unknown output, the probabilistic classifier model is to be is built over the variables x, y and z with a graphical model similar to that of [Yan et al., 2010b], as shown in Figure 6.3. Nevertheless, the joint conditional probability can be represented as:

$$p(Y, Z|X) = \prod_i p(z_i|x_i) \prod_t p(y^{(i)}|x_i, z_i)$$

This is the base of the simulation model to be implemented, however, there are some assumptions and conditions to be detailed. The sections below will extend details of the formalisation.

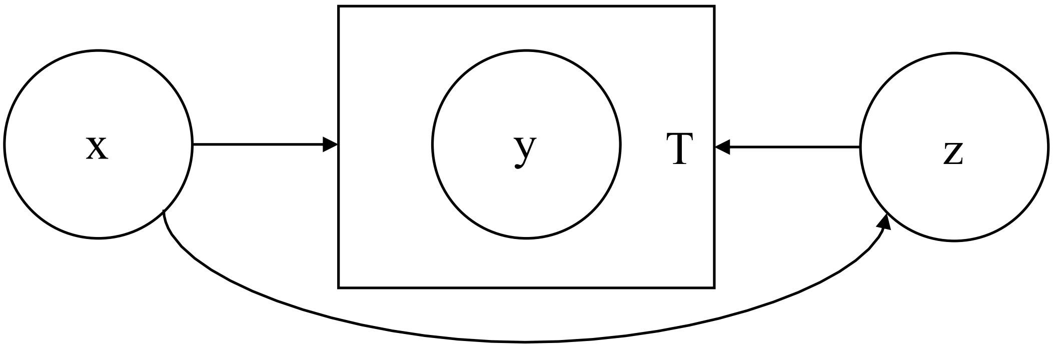


Fig. 6.3 Graphical Model for x, y, and z.

6.6.2 Assumptions

It is assumed that annotation provided by annotator t is dependent on both the input x and the unknown output z. It is assumed that annotators do not all have the same expertise so that all are equally good or bad, so when annotating data, the quality of annotation depends on the observed input. It is also assumed that all annotators $t = \{1, \ldots, T\}$ are independent when observing an input and providing a label.

Moreover, to complete the model, a definition of the form of the conditional probability is given in the next section.

6.6.3 Conditional destitution

In the utilised model, the assumption is that annotators provide a noisy version of the true label z. A function indicated as θ is generated, that $\theta^x \to z^x + \delta$, where δ is an additive noise. Now if annotator t is an expert and observes the sample x_i, he or she should be able to give a label $y_i^{(t)}$ that is as good as z^i. Therefore, if all samples are labelled by experts, we should have $\mathcal{Y} \equiv \mathcal{Z}$.

This assumption corresponds to [Raykar et al., 2010] in the cases where there are no gold-standard answers.

Part III

Experimental Framework

Chapter 7

Active Expert Sourcing for Digital Humanities

It is assumed that the comprehensive literature review in the previous chapters has set the scene for the reader.

The chapters 2 and 3 discuss the importance of domain-specific NER. Then, two chapters discuss the methods used to develop domain-specific named entity recognition (NER), namely, active learning and crowdsourcing, and two chapters are dedicated to these techniques, Chapters 5 and 4, respectively.

This thesis describes the results of adapting some of the techniques presented in the literature review and adds a number of contributions, which are presented earlier in Chapter 6.

Next, the hypotheses that this thesis posits are tested in this chapter and in the following chapter, (Chapter 8). This chapter describes a real-world experiment to test the feasibility of the proposed approach to knowledge extraction for domain specific information, more specifically, domain-specific NER.

The first experiment was done within the archaeology domain. The aim was to create a real-world implementation of the proposed approach and report the findings.

Specifically, a close collaboration with experts in the archaeology domain was established. The team behind the annotation schema for the archaeological text [Bonin et al., 2012] and the team that built an active learning attempt [Ekbal et al., 2012a] were both contacted. Also, the archaeologists who worked in the The Humanities Research Portal or The *Portale della Ricerca Umanistica in Trentino*[1] (PRU) project were approached to participate. The key contacts among the domain experts were archaeologists who were part of the annotation schema.

Also, a web portal was developed and used in this experiment. This web portal allows experts to upload their publications, which are then processed and the results are displayed as plain text. However, the display page highlights a number of tokens for experts to annotate. These highlighted instances are selected by an active learning (AL) algorithm. The experts provide feedback to the AL algorithm, and then the loop starts again.

The implementation of the proposed approach included a number of stages, which are discussed below.

In relation to the research objectives detailed in Section 6.2, the experiments in this chapter try to show the feasibility of creating an online end-to-end expert-in-the-loop framework for annotating domain-specific named entities (e.g. hypothesis 0.1), and how such platforms can produce high-quality annotated data (e.g. hypothesis 0.2). The experiments show how boundaries can be pushed with limited resources (e.g. hypothesis 0.3 & 0.4) and how experts can be easily be found if approached in the right way and place (e.g. hypothesis 0.5). The experiments also show that experts are likely to annotate their own text if asked to do so (e.g. hypothesis 0.6), especially if the tasks they are asked to do is kept to the minimum due to the use of Al (e.g. hypothesis 0.7).

[1] http://apsat.mpasol.it/

Chapter structure

This chapter describes the real-world implementation of the proposed approach: *active expert sourcing*. The chapter begins by discussing the reproduction of a prior experiment on the use of AL in the archaeology domain at the first part. Then the setting of the experiment is presented. Later, the results of this research are presented and discussed at the second part of this chapter.

7.1 Reproducing an archaeology AL NER

The first experiment completed as part of this research reproduced the active learning experiment found in Ekbal et al. [2012a].

Ekbal et al. [2012a] uses conditional random fields (CRF) for NER [Lafferty et al., 2001]. CRFs are conditionally trained probabilistic finite states, which in this case are an undirected graphical model. CRFs are a very common algorithm used to build NERs as it can incorporate a large number of non-independent features while still being an efficient producer of non-greedy finite-state inferences. In NER, a CRF algorithm would observe a sequence of tokens and then produce a sequence of labels. The sequence of tokens can be either a sentence or a document.

The following section details the system of Ekbal et al. [2012a] followed by this researcher's reproduction of the system.

7.1.1 Ekbal et al. [2012a] AL implementation

A common implementation of CRF is software called `CRF++`. Ekbal et al. [2012a] used `CRF++` version 0.54 `CRF++` and achieved high scores in a number of shared The SIGNLL Conference on Computational Natural Language Learning (CoNLL) tasks.

The software is open source and can be obtained without registration or having to email the developers.

`CRF++` used the training data-set; each line of data represented a vector of features. The data-set was in tab format, similar to the data-sets provided by shared CoNLL tasks. Each line represented a token or a set of features, and the final column is the target tag/label.

The feature set used for each word consisted of the following:

- Root.
- Prefix.
- Suffix.
- PoS.
- Previews of the word.
- Frequency of the word.
- Content words.
- Capitalization.
- If the word contains digits.

A number of setting parameters were used. Table 7.1 shows these parameters.

The paragraph below details each of the features which were extracted, represented and used.

Feature set

1. Context word: The surrounding words of each token are taken into account (e.g. the preceding and succeeding words).

Table 7.1 Parameter setting for `CRF++`

Parameters	Value	Note
a	L2	Regularization parameter.
c	1	Soft-margin parameter, which determines the balance between overfitting and underfitting.
f	1	Features that occur at least once in the given training data.

2. Word suffix and prefix: There is a simple fixed-length prefix and a suffix, which was set to 3. However, if the currently observed token/word is shorter than three characters, this feature is not used.

3. First position: This is binary value is used to denote whether this observed token is the first word of a sequence/sentence.

4. Word length: This is a binary value that represents if the observed token is longer than 5 characters. The reason for doing this is that short words are not likely to be named entities (NEs).

5. Infrequent word: This is a binary value denoting whether a token is a very frequently used word that appears in the data-set. The reason for this is that very frequently used words are not likely to be NEs.

6. Capitalization: A binary value denoting if the word is capitalized.

7. Part-of-Speech (PoS) tag.

8. Root word: The stem of the word.

Since the feature set for training was ready, the paragraphs below will give details about the data-set.

Data-set

The data-set used in this experiment consists of 11 articles from the archaeology domain. The articles were mostly obtained from the journal *Preistoria Alpina.* The data-set consisted of about 50,000 tokens which were annotated by domain experts (Table 7.2).

Table 7.2 Ekbal et al. [2012a] data-sets

Data-set	Tokens	NEs
Training	20,739	2,611
Developing	5,292	622
Testing	11,534	1,582

The NE annotation schema used was the one presented in the literature review of this thesis at Table 3.3. This data-set was then used to implement AL, as discussed below.

Active learning algorithm

The algorithm used here is similar to the one mentioned in the literature review at Algorithm 1. However, in this section, a more detailed version is presented in Algorithm 1

Ekbal et al. [2012a] results

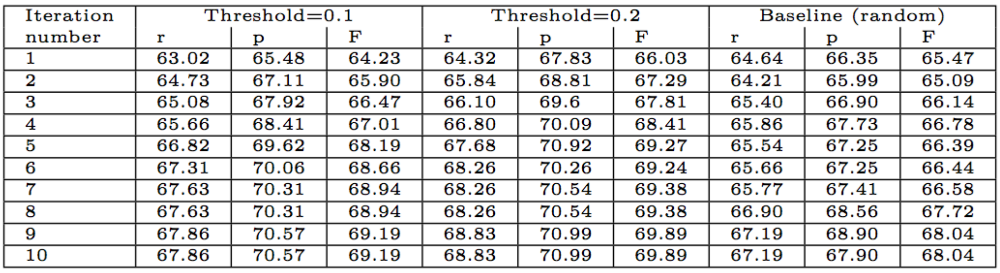
Ekbal et al. [2012a] reported the following results in their paper (Figure 7.1):

7.1.2 Reproducing Ekbal et al. [2012a]

's experiment This section offers a detailed description of this researcher's attempt to reproduce Ekbal et al. [2012a]'s experiment. Next, this paper gives a detailed description of the procedure that was followed, and this section concludes by presenting the results of this researcher's experiment.

Algorithm 2 Ekbal et al. [2012a] active learning algorithm

```
evaluates the system according to the gold standard test data.
tests the development data and calculate the conditional probabilities of all output classes.
computes the confidence interval (CI) between the two most probable classes for each token.
If the CI is below the threshold value (set to 0.1 and 0.2), then add the NE token, along with its sentence identifier and CI in a list of effective sentences selected for active annotation (named EA).
sorts EA in ascending order of CI.
selects the top 10 sentences.
removes the top 10 sentences along with the preceding and following sentences from the development set.
adds the sentence to the training set.
retrains the CRF classifier and evaluate with the test set.
Repeats the process.
```



Iteration	Threshold=0.1			Threshold=0.2			Baseline (random)		
number	r	p	F	r	p	F	r	p	F
1	63.02	65.48	64.23	64.32	67.83	66.03	64.64	66.35	65.47
2	64.73	67.11	65.90	65.84	68.81	67.29	64.21	65.99	65.09
3	65.08	67.92	66.47	66.10	69.6	67.81	65.40	66.90	66.14
4	65.66	68.41	67.01	66.80	70.09	68.41	65.86	67.73	66.78
5	66.82	69.62	68.19	67.68	70.92	69.27	65.54	67.25	66.39
6	67.31	70.06	68.66	68.26	70.26	69.24	65.66	67.25	66.44
7	67.63	70.31	68.94	68.26	70.54	69.38	65.77	67.41	66.58
8	67.63	70.31	68.94	68.26	70.54	69.38	66.90	68.56	67.72
9	67.86	70.57	69.19	68.83	70.99	69.89	67.19	68.90	68.04
10	67.86	70.57	69.19	68.83	70.99	69.89	67.19	67.90	68.04

Fig. 7.1 Evaluation results of AL with threshold 0.1 and 0.2 compared to random selection.

Description of software developed

The first part of implementing the system is to develop an NER that would later be modified to allow for an AL cycle to be implemented. Therefore, the first step was to build a CRF based NER.

The NER required training data to train the NER and a test set for evaluation of the NER. A development set could also have been included when developing the feature set. However, since the aim of this part of the experiment was to reproduce an already published system, no development set was needed at this stage.

The researcher acquired the same data-set used in [Ekbal et al., 2012a]. It was found that the data-set was already divided into training, developing and testing sets. Therefore, the data-set was kept intact. The data-set was in tab separated volume (tsv) format.

The first column is for tokens/words from sentences, and each sentence is separated by an empty token.

The following columns are features extracted from the full text that the data-set comes from. The columns of the data-set conform to the list of the feature set discussed earlier.

The tool `CRF++` was downloaded from its developer's website using a computer running the Linux operating system.

The course code is available, so it was also downloaded and built on the machine. This tool is a conditional random field (CRF) implementation in `C++`, hence its name. The format of training data that `CRF++` accepts is of CoNLL format, which is in `tsv`. As training data was already provided in `tsv` no preprocessing was necessary. To let `CRF++` know about the feature set that it would learn, a file containing a "template" was fed to the tool. This template tells the tool which feature to learn from the model.

At this stage the NER model was ready to be trained, so the training set was given to `CRF++` along with the template file. Then `CRF++` learned one of the models. The training process is time consuming and depends on both the size of the training data-set and the complexity of the feature set (e.g. template).

After a model is learned, one can use `CRF++` to annotate/tag/classify new data.

To evaluate the model, `CRF++` uses the learned model to tag the test set. After tagging the test set the model can be evaluated.

To evaluate the model and report the results, a `Perl` script provided by CoNLL was used. The script was available to download from CoNLL shared tasks websites.

The script includes a `tsv` in which the first column is a token and the last two columns are the predicted and the gold class/tag.

In the first implementation of the NER only the training data-set was used to learn the model.

In the second implementation both the training and developing set were combined to learn a model. The results are detailed in Table 7.3.

Table 7.3 Initial NER result

	Recall	Precession	F-score
Training set	63.33	66.44	64.85
Training + developing sets	67.87	71.41	69.60

At this point it was clear that the NER system was working and was able to incorporate the same data-set and feature set. Therefore, the next step was to detail the procedure to be followed to re-implement the [Ekbal et al., 2012a] algorithm, 1.

Procedure followed

The algorithm for AL started with an initial training data-set that began the AL cycle. In [Ekbal et al., 2012a] the training set was used as the initial data needed to learn the AL model. Then the trained model was used to tag the development set.

Since `CRF++` can check for all possible tags for a given token, it can also output the conditional probabilities of all other possible classes/tags. This output was used to calculate the conditional interval of the two most probable classes for each token.

If the calculated conditional interval was below the threshold value of 0.1, and later 0.2, the sentence in which the NE/token appeared was selected as potential data to be added to the training set. This potential data was then sorted in ascending order according to its conditional interval and the top 10 sentences were added to the training set. It is worth noting that in addition to these 10 sentences the previous and the following sentence of each of the top 10 sentences were also added to the training set. So, in total, 30 sentences were added to the training set.

This was the end of an AL cycle. As a new, updated training set became available a new model was trained using this data.

Results

Two AL experiments are described in this section. One experiment used a threshold of 0.1 to select the most informative instances, and the other experiment used an increased threshold of 0.2. The results found in these experiments are close enough to the results reported in [Ekbal et al., 2012a]. Which suggest that the data is valid, and that the training sets are working correctly. The full results of both experiments are shown in Table 7.4 and compared to the results of random selection. Also the results are presented in Figure 7.2.

Table 7.4 Ekbal et al. [2012a] reproduction results

	Threshold 0.1			Threshold 0.2			Random		
Iteration	R	P	F	R	P	F	R	P	F
1	63.81	65.27	64.53	63.94	65.06	64.50	63.71	65.77	64.72
2	64.66	66.90	65.76	64.50	65.03	64.76	64.20	67.11	65.62
3	65.00	66.68	65.83	64.99	65.96	65.48	64.54	65.63	65.08
4	65.55	68.89	67.18	65.73	66.96	66.34	65.43	66.75	66.08
5	65.76	69.54	67.60	65.97	66.89	66.43	65.50	68.45	66.94
6	65.92	67.64	66.77	66.41	68.61	67.49	66.39	68.10	67.23
7	66.87	68.20	67.53	66.89	69.35	68.10	66.89	69.51	68.17
8	67.44	70.13	68.76	67.36	69.36	68.35	66.94	68.30	67.62
9	67.65	71.07	69.32	68.20	70.74	69.44	67.08	68.94	68.00
10	67.83	70.27	69.03	68.73	70.96	69.83	67.18	68.94	68.05

The next section will detail the experiments conducted to test the hypothesis of this thesis. These experiments were performed in the archaeological domain.

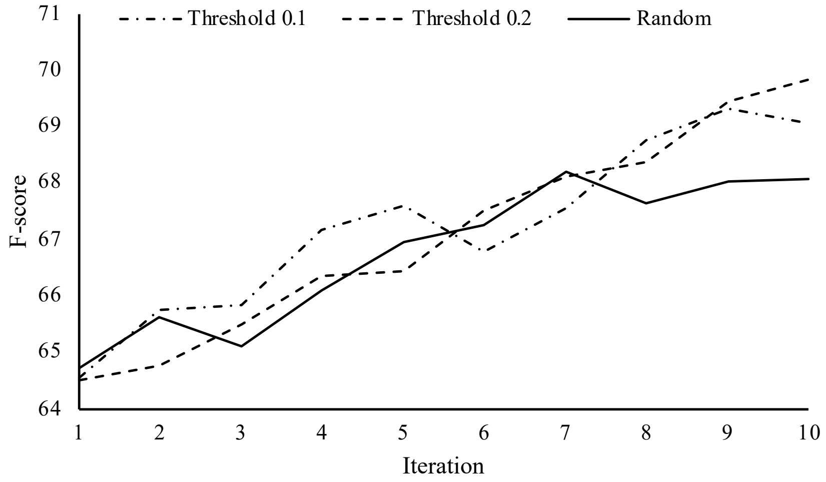


Fig. 7.2 Reproducing Ekbal et al. [2012a] results.

7.2 Active expert sourcing for archaeology

In this experiment, a new variant of active learning is proposed which is henceforth called **active expert sourcing**. The researcher's proposed approach combines active learning with crowdsourcing in addition to the feedback required by active learning, which was provided by the same experts who produced the research papers being processed for this study.

The framework developed will then be integrated in the Apsat/Alpinet GIS portal, The Humanities Research Portal or The Portale della Ricerca Umanistica in Trentino[2] (PRU).

When a scholar uploaded a paper to the PRU portal, an NLP pipeline including a named entity recogniser that had initially been trained on a small amount of manually annotated data was used to extract information. The scholar was then asked to provide feedback on some NEs that the NER was less certain about. This feedback was then used to retrain the NER.

[2] http://apsat.mpasol.it

The following sections will add more details regarding the software and methods used in the experiments and the results of the experiments.

7.2.1 Description of the software developed

This section details the software developed and used for the experiments in this part of the thesis.

A full NLP pipeline was developed. The pipeline developed consisted of several modules which will be explained in detail below.

The data came in the `PDF` format. The first module extracted text from the w`PDF`s. This was done using `pdf2txt`. Unfortunately, the text was not always perfectly extracted from `PDF`. Therefore, to check if all the necessary text had been extracted, a comparison between the number of pages extracted by `pdf2txt` and that of `pdfinfo` was made. If the number of pages was not identical further examination of the file was needed. Also, if the total number of pages was less than 2 the file was ignored.

The second stage was to check the language of the document/text. The pipeline can handle both English and Italian text. So determining the language at an early stage was necessary. For this, the `CLD` Python package was used. This tool is a binding around Google Chromium's embedded compact language detection library.

The third module in the pipeline which was designed to extract the different sections/parts of the text. So, as in the original `PDF`, there were different sections (e.g. title, authors, abstract, introduction, citation, etc.); in this step the pipeline tried to extract such sections as those mentioned above. The extraction of the sections was done using `ParsCit`, which uses supervised machine learning to detect and classify section of scientific papers. Two models of `ParsCit` were used, one for English and one for Italian. So, depending on the language detected in the previous step, the relevant model was run.

The fourth module in the pipeline was developed using an NLP software called `TextPro`. Although this software is free, it required the user registration to download. There are many components in `TextPro`, including:

- Tokenization.
- Sentence segmentation.
- Part of Speech (POS) tagging.
- Lemmatisation.
- Basic NER.

For each of these steps `TextPro` has an optimised model for different languages including both English and Italian, and this was helpful for the two languages of text in this experiment. This feature was utilised in the pipeline the researcher developed. Since the documents being processed contain either English or Italian, and some document were written in both English and Italian, in this step of the pipeline, each section was processed using the appropriate language settings. For example, if a paper or document's abstract is written in Italian and the rest of the document is in English, the abstract was processed independently from the rest of the document. In other words, the abstract of that document was processed using TextPro's Italian PoS, NER, etc. tools and the rest of the document was processed using the English version of TextPro.

Next is the fifth module in the pipeline which prepares the rest of the feature set for the AL module. This step involves an AL NER, which is the same software described in 7.1 and was developed for reproducing [Ekbal et al., 2012a] as part of this thesis.

7.2.2 Method and procedure followed

The authors of the archaeology schema [Bonin et al., 2012], which was utilised though out the experiments in this chapter, were contacted and of six archaeologists participated in the experiments discussed in this section. These expert archaeologists were two professors and four of their post-graduate students/candidates.

An online website was created that allows users to log in and upload their publications. The aim was to integrate this website into the Apsat/Alpinet GIS portal PRU[3] However, due to technical and time constraints the website was developed as a standalone system with the aim of integrating it back to the portal during future work. Users were able to upload their published papers, which triggers the processing pipeline.

The uploaded paper is then processed to extract information, which is then passed to the classifier to extract NEs. The classifier adapts an AL approach by asking for feedback whenever it is unsure how to classify an entity.

The model developed in the previous experiment, (Section 7.1),was used as a starting point for the AL algorithm. Then, after getting feedback from users the model was retrained.

Each expert user had a unique username and password. The expert was able to log in and upload their paper; the paper was processed and then shown to them in text format with some words/tokens highlighted in a red colour. These highlighted items are the tokens the AL NER flagged as the most uncertain items. The method used to flag such tokens is the same AL algorithm described in the previous sections.

The experts were then asked to provide feedback on highlighted each token. This feedback was added to the training set according to the AL algorithm described earlier.

[3] .

Then the model was retrained. According to the assumption of AL, the model should then improve as a result of integrating the experts' feedback.

As discussed in the research methodology section (Section 6.4 of Chapter 6), four settings were developed –which are also mentioned below. The reason for developing these settings was to have a baseline to compare the proposed approach to. Since the proposed approach of this thesis consists of different components it was thought that having multiple baselines consisting of different settings would give the researcher a better understanding of the findings.

In total, four experimental settings were used in the next stage of the research: Active Expert Sourcing (+A+E), Expert Sourcing (−A+E), Active Crowdsourcing (+A−E) and Crowdsourcing (−A−E). In other words, there were four experiments which were to be run simultaneously. All the experiments were conducted on the same website and will be explained in detail below..

The pipeline developed by the researcher has been discussed above. However, since there are four settings in this experiment, a separate pipeline was assigned to each setting. Thus, every experimental setting had a distinct pipeline which resulted in a distinct AL NER model. Below, each setting is evaluated and the results are reported. Also, all the settings were evaluated and compared.

7.2.3 AL with CRF based NER

This section introduces the supervised machine learning approach used to develop the NER system based on CRFs, and the approach used to select the most informative samples as adopted in this work.

CRFs [Lafferty et al., 2001] are one of the dominant paradigms used to train models for NER. CRFs calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence $s =< s_1, s_2, \ldots, s_T >$ given an observation sequence of $o =< o_1, o_2, \ldots, o_T >$ is calculated as follows:

$$P_{\wedge}(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

where $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight, λ_k, is to be learned via training. The values of the feature functions may range between $-\infty, \ldots and + \infty$, but they are typically binary.

When applying CRFs to the NER problem, an observation sequence is a token of a sentence or document of a text, and the state sequence is its corresponding label sequence.

Next, the C++ based CRF++ package [4] was use; it is a simple, customizable, and open-source implementation of CRF for segmenting or labelling sequential data.

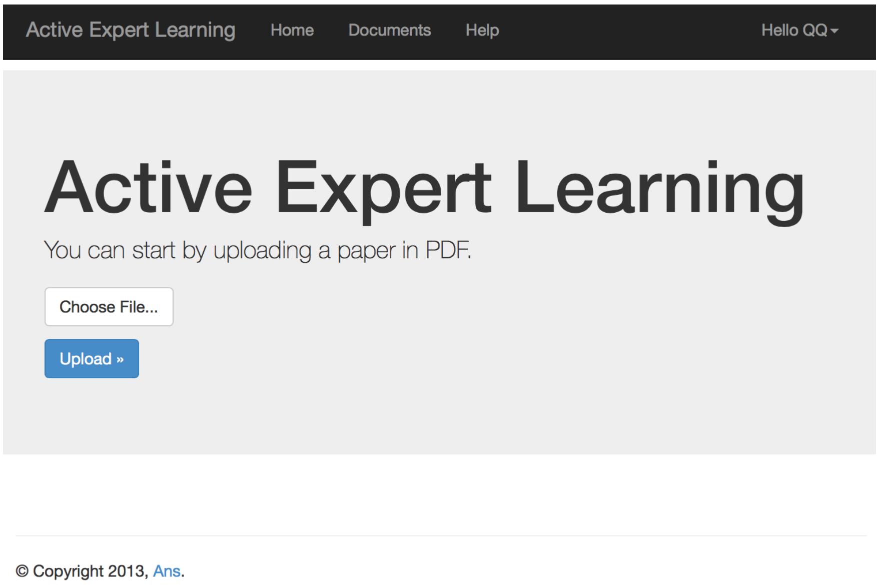
7.2.4 Implementation of online system

In order to compare these conditions, a website (Figure 7.3) was created in which users, after using the login page (Figure7.4), were presented with documents on which to give feedback (Figure 7.5). Each user was classified into one of two categories: expert or non-expert. Please note that all screenshots are taken from this website.

On the document view page, the users (both experts and non-experts) see a full-text view of the processed document. Also, the users can see the entities identified by the website (the greyed-out words).

In Figure 7.6, when a user hovers over a greyed-out word, a tool-tip shows the class assigned by the NER. Moreover, the elements highlighted in red are entities that need feedback. Again, the class assigned can be revealed by hovering over them as in Figure 7.7.

[4] https://taku910.github.io/crfpp/



Active Expert Learning

You can start by uploading a paper in PDF.

© Copyright 2013, Ans.

Fig. 7.3 A screen-shot of the online website developed for the experiment on named entities for the archaeological domain. Please note that the early version of the system was called Active Expert Learning

After a user clicks on any of the flagged entities, it is then highlighted in green to indicate that it has been selected. The user can expand the boundaries of the selected entity on either side, as shown in Figure 7.8.

When an entity has been selected for annotation a list of available classes is revealed on the left-hand side (Figure 7.9a). In this list, the users can correct an entity's class or confirm that the current tag is correct. Moreover, for more enthusiastic users, a sub-list is available to tag entities according to the BIO format, as shown in Figure 7.9b.

The documents which the users gave their feedback were processed using four distinct versions of the pipeline. Initially, each pipeline included a NER system based on the CRF model described in the previous section, which was trained with the available training/seed data. Then, the users provided feedback for new documents. For this study, each user was asked to provide feedback on 50 items, as shown in [Ekbal et al., 2011].

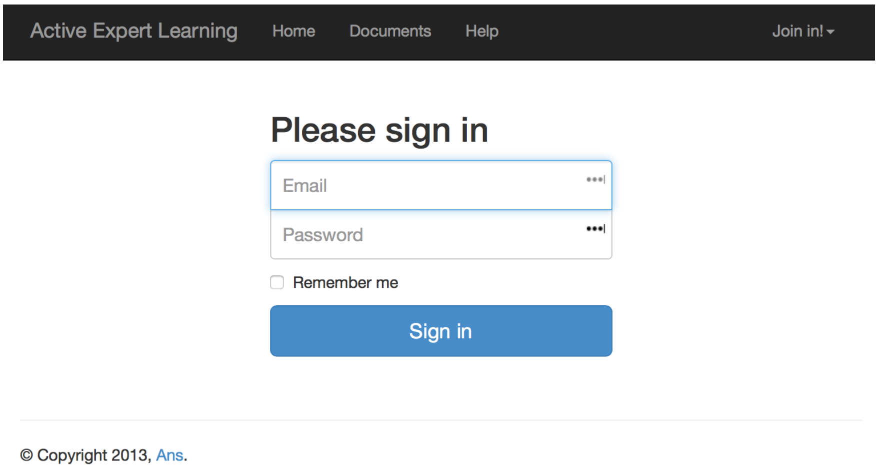


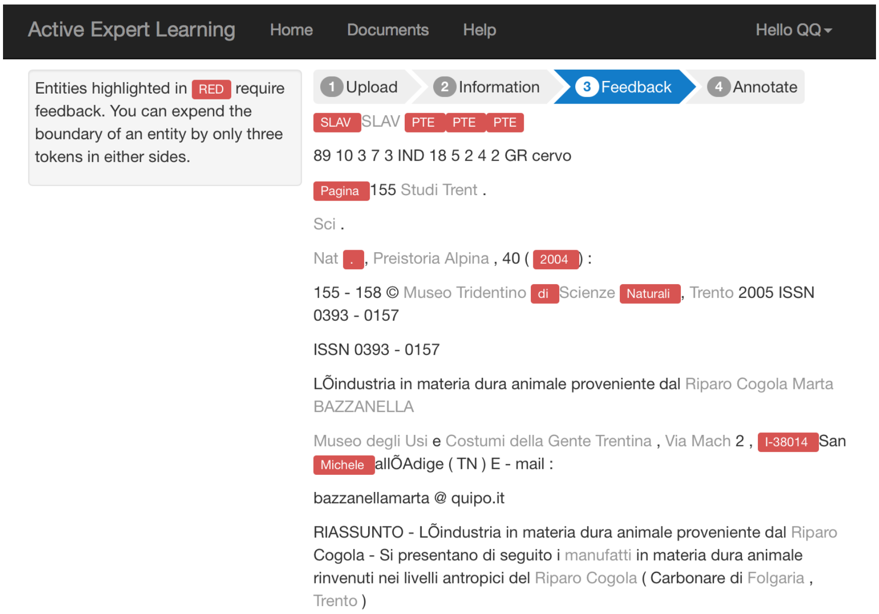
Fig. 7.4 A screenshot of the login page for the online website developed for the experiment on named entities for the archaeological domain

As the feedback was collected, the models were retrained overnight.

7.2.5 Results

This section reports the results of the four experimental settings used in this experiment: *active expert sourcing* (+A+E), *expert sourcing* (−A+E), *active crowdsourcing* (+A−E) and *crowdsourcing* (−A−E). Figure 7.10 shows the F-scores reported for every iteration of the AL cycle. The results show that the method proposed in this thesis outperformed the other settings. Specifically, the proposed approach's performance was significant with the $p - value < 0.05$ as compared to other baseline settings. Table 7.5 presents the F-score for each iteration of each setting developed as part of this experiment, along with the number of instances and sentences added at each iteration.

However, there were no significant differences in all of the iterations as no significant differences between settings were found until iteration six. Hence, after the sixth iteration, each experimental setting was tested against the other settings. Table 7.6 shows the $p - value$ of the Friedman test and Wilcoxon tests.



Entities highlighted in RED require feedback. You can expend the boundary of an entity by only three tokens in either sides.

SLAV SLAV PTE PTE PTE

89 10 3 7 3 IND 18 5 2 4 2 GR cervo

Pagina 155 Studi Trent .

Sci .

Nat . , Preistoria Alpina , 40 (2004) :

155 - 158 © Museo Tridentino di Scienze Naturali , Trento 2005 ISSN 0393 - 0157

ISSN 0393 - 0157

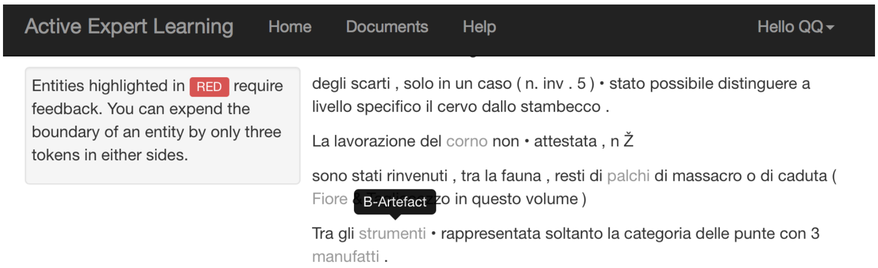
LÕindustria in materia dura animale proveniente dal Riparo Cogola Marta BAZZANELLA

Museo degli Usi e Costumi della Gente Trentina , Via Mach 2 , I-38014 San Michele allÕAdige (TN) E - mail :

bazzanellamarta @ quipo.it

RIASSUNTO - LÕindustria in materia dura animale proveniente dal Riparo Cogola - Si presentano di seguito i manufatti in materia dura animale rinvenuti nei livelli antropici del Riparo Cogola (Carbonare di Folgaria , Trento)

Fig. 7.5 A screenshot of the full text of a document uploaded by an expert



Entities highlighted in RED require feedback. You can expend the boundary of an entity by only three tokens in either sides.

degli scarti , solo in un caso (n. inv . 5) • stato possibile distinguere a livello specifico il cervo dallo stambecco .

La lavorazione del corno non • attestata , n Ž

sono stati rinvenuti , tra la fauna , resti di palchi di massacro o di caduta (Fiore zo in questo volume)

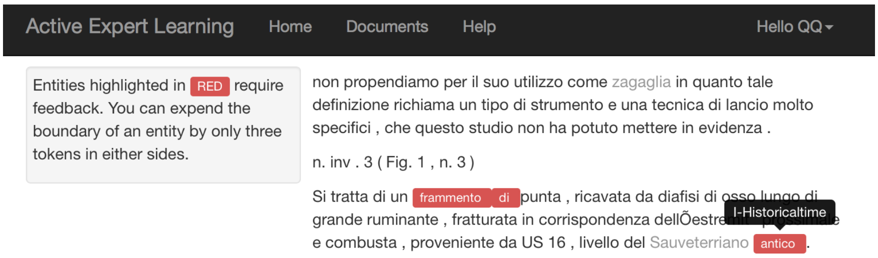
Tra gli strumenti • rappresentata soltanto la categoria delle punte con 3 manufatti .

Fig. 7.6 A screenshot of showing an entity and the class assigned to it

Table 7.7 shows the total number of entities tagged by each setting. However, table shows the number of instances added to training at each iteration. As shown in the table, crowdsourcing contributed more annotations than any other setting.

7.2.6 Developed data-set

Based on the number of tagged entities, the experiments managed to collect an adequate amount of data in each of the four settings. This data is significant when it comes to data-set curating, especially for domains such as the digital humanities—especially when considering that this data was collected using limited resources. Active learning guided



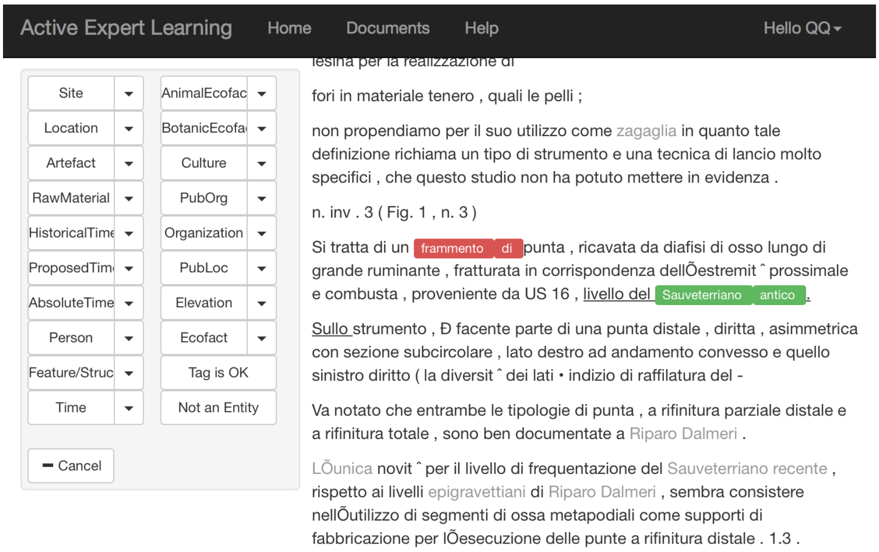
Entities highlighted in RED require feedback. You can expend the boundary of an entity by only three tokens in either sides.

non propendiamo per il suo utilizzo come zagaglia in quanto tale definizione richiama un tipo di strumento e una tecnica di lancio molto specifici , che questo studio non ha potuto mettere in evidenza .

n. inv . 3 (Fig. 1 , n. 3)

Si tratta di un frammento di punta , ricavata da diafisi di osso lungo di grande ruminante , fratturata in corrispondenza dellÕestre e combusta , proveniente da US 16 , livello del Sauveterriano antico .

Fig. 7.7 A screenshot of an entity highlighted in red that is to be annotated by an expert on the website. The current tag for this entity can be revealed by hovering over the entity.



lesina per la realizzazione di

fori in materiale tenero , quali le pelli ;

non propendiamo per il suo utilizzo come zagaglia in quanto tale definizione richiama un tipo di strumento e una tecnica di lancio molto specifici , che questo studio non ha potuto mettere in evidenza .

n. inv . 3 (Fig. 1 , n. 3)

Si tratta di un frammento di punta , ricavata da diafisi di osso lungo di grande ruminante , fratturata in corrispondenza dellÕestremit ˆ prossimale e combusta , proveniente da US 16 , livello del Sauveterriano antico .

Sullo strumento , Ð facente parte di una punta distale , diritta , asimmetrica con sezione subcircolare , lato destro ad andamento convesso e quello sinistro diritto (la diversit ˆ dei lati • indizio di raffilatura del -

Va notato che entrambe le tipologie di punta , a rifinitura parziale distale e a rifinitura totale , sono ben documentate a Riparo Dalmeri .

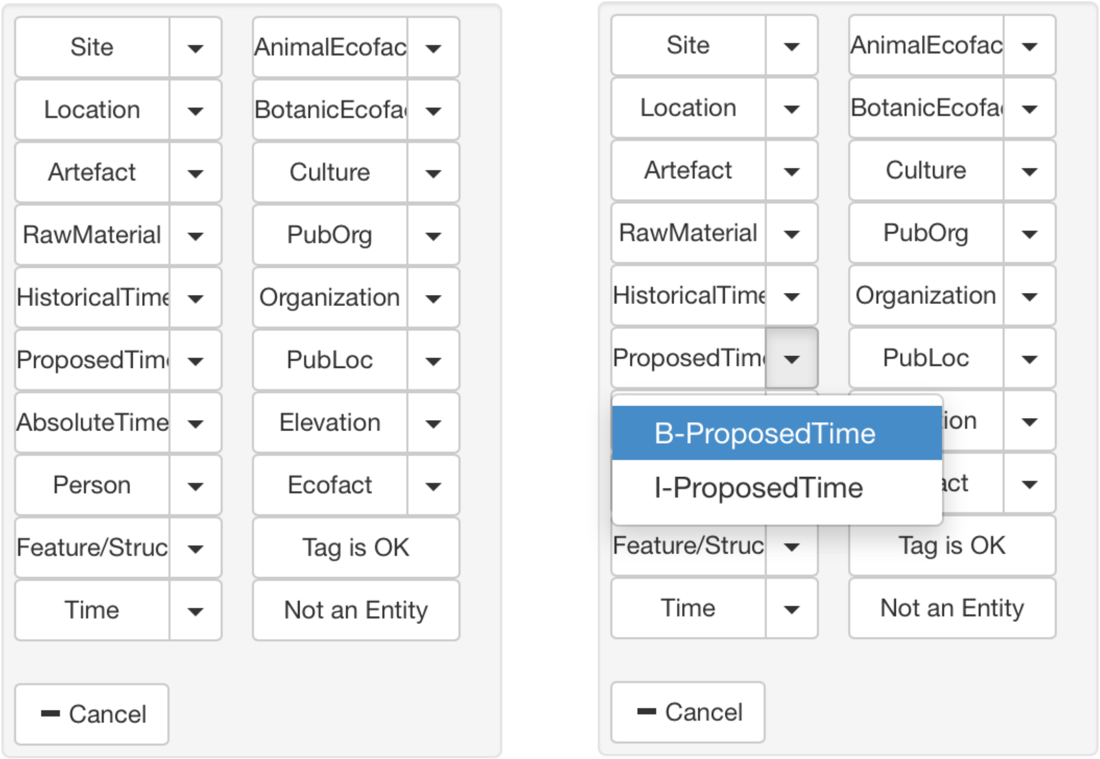
LÕunica novit ˆ per il livello di frequentazione del Sauveterriano recente , rispetto ai livelli epigravettiani di Riparo Dalmeri , sembra consistere nellÕutilizzo di segmenti di ossa metapodiali come supporti di fabbricazione per lÕesecuzione delle punte a rifinitura distale . 1.3 .

Fig. 7.8 A screenshot of an entity selected for annotation.

the crowd, whether they were experts or not, toward the creation of a high-quality data-set.

A sample of an annotated text is shown in Table 7.10 and in Table 7.11. In total, 28 documents were annotated by experts. Table 7.8 show details about the data-set developed that was annotated by experts in the setting *active expert sourcing*. Also Table 7.9 details the topics/NE types included compared to the schema at Table 3.3.

However, there was no cross-validation as no two experts annotated the same document; therefore it is not possible to carry out inter-rater reliability test such as those for Cohen's kappa, Fleiss' kappa, or Krippendorff's alpha. This issue is



(a) List of classes that can be used to annotate an entity.

(b) The sublist that can be used for annotating an entity according to BIO format.

Fig. 7.9 A screenshot of the list of classes that can be used to annotate an entity.

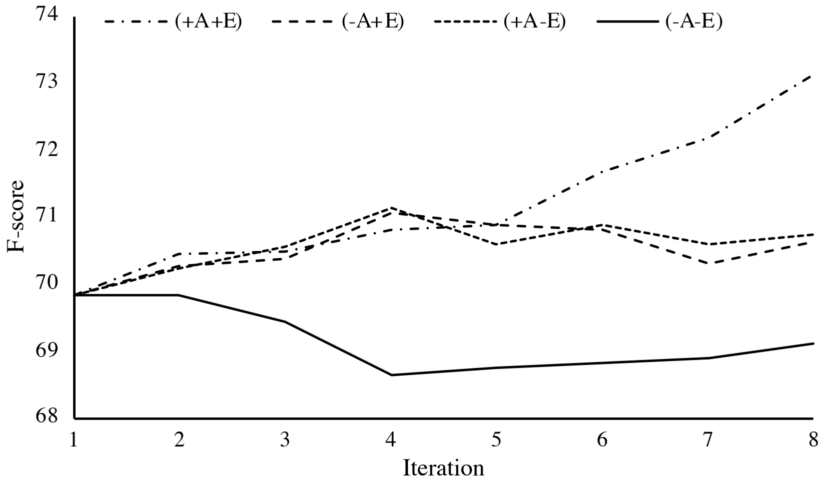


Fig. 7.10 The F-scores reported for the experiment on the archaeological domain.

Table 7.5 Number of instances/sentences added at each iteration in archaeology experiments, along with the F-score

Iteration	Sentences	+A+E	-A+E	+A-E	-A-E
1	0	69.83	69.83	69.83	69.83
2	158	70.45	70.25	70.22	69.83
3	95	70.48	70.35	70.56	69.42
4	51	70.79	71.06	71.13	68.62
5	43	70.87	70.87	70.57	68.73
6	108	71.68	70.8	70.87	68.8
7	91	72.18	70.29	70.57	68.88
8	7	73.11	70.62	70.72	69.11

considered to be among the limitations of this thesis. Furthermore, not being able to perform inter-rater reliability tests may not be helpful for addressing Hypothesis 0.2 which is: *Approaching different platforms to current crowdsourcing platforms will yield high-quality annotated data for domain-specific NER.*

7.2.7 Discussion

The experiments in this section of this chapter concerned a set of four settings; one is the researcher's proposed and novel approach and the other three are baselines.

Table 7.6 Significance analysis of experiments on archaeology

Iteration	Friedman Test	+A+E vs. -A+E	+A+E vs. +A-E	+A+E vs. -A-E	-A+E vs. +A-E	-A+E vs. -A-E	+A-E vs. -A-E
1	0.280	0.159	0.219	0.066	0.344	0.422	0.479
2	0.465	0.062	0.324	0.128	0.283	0.430	0.050
3	0.476	0.382	0.246	0.006	0.487	0.398	0.341
4	0.444	0.447	0.464	0.023	0.466	0.012	0.029
5	0.023	0.059	0.260	0.037	0.373	0.018	0.047
6	0.036	0.047	0.039	0.032	0.310	0.026	0.039
7	0.464	0.017	0.006	0.004	0.398	0.021	0.029
8	0.025	0.017	0.013	0.032	0.476	0.011	0.045

Table 7.7 Feedback collected for entities obtained from each setting in archaeology

Setting	Total number of entities
+A+E	370
-A+E	710
+A-E	312
-A-E	894

The reason for having three baselines is that the author's proposed approach includes different components; by making comparisons with the additional baselines all these components can be studied.

The results pertaining to each setting are presented in individual sections below; however, there are many cross-references to other settings as needed.

Crowdsourcing (−A−E)

The setting *crowdsourcing* achieved a noticeable increased performance, however, it was lower than all other settings in term of F-score. The results also showed that the

Table 7.8 Details of the developed data-set.

	English	Italian	Total
Number of documents	6	22	28
Number of tokens	73195	176315	249510
Number of entity types	15	18	18
Number of entity tagged	163	693	856
Number of tokens tagged	319	1345	1664

crowdsourced participants were willing to provide a considerable amount of feedback, even when not guided by active learning.

Moreover, having more feedback from crowdsourcing also meant that even when there was noise in the participants' feedback, the performance still managed to improve slightly, which suggests that there is a chance that the amount of helpful feedback outnumbered unhelpful feedback over time.

More feedback was collected by using *crowdsourcing* than with any other method. Perhaps the level of freedom allowed in tagging in this setting contributed to the higher amount of feedback received.

However, more feedback did not translate into better NER performance.

Expert sourcing (−A+E)

The freedom of choice can also be clearly seen in the setting *expert sourcing* (−A+E). Crowdsourced experts were given the freedom to annotate whatever they thought was worthy of annotating. The number of annotations they provided suggests that experts were willing to take part. The experts performed the assigned task in a crowdsourcing fashion as explained in the literature review.

Moreover, *expert sourcing* was able to make decisions about what words to tag and what words not to tag. This setting resulted in less feedback compared to non-experts. However, *expert sourcing* resulted also in much lower amount of noise, which is evident

Table 7.9 Details of topics/NE types included in the data-set.

NE type	English	Italian
Absolutetime	✓	✓
Animalecofact	✓	✓
Artefact	✓	✓
Botanicecofact	✗	✓
Coordalt	✓	✓
Culture	✓	✓
Ecofact	✗	✓
Feature	✓	✓
Historicaltime	✓	✓
Location	✓	✓
Material	✓	✓
Organization	✓	✓
Person	✓	✓
Proposedtime	✓	✓
Publoc	✓	✓
Puborg	✓	✓
Site	✓	✓
Time	✗	✓

from its F-scores. Experts who used this setting in some stages/iterations performed even better than when they were guided with active learning (e.g. *active expert sourcing* +A+E).

However, as compared to *active expert sourcing*, experts using the *expert sourcing* setting provided more feedback than necessary to achieve a higher performance. In other words, the performance of *expert sourcing* can be achieved with only 50% of the effort needed by expert annotators as compared to the *active expert sourcing* setting (+A+E). This was also the case when comparing *active crowdsourcing* (+A−E) to *crowdsourcing* (−A−E).

Active expert sourcing (+A+E)

As shown by the results in (Figure 7.10), active expert sourcing managed to significantly outperform the other settings ($p - value < 0.5$). Although the proposed method struggled slightly at the beginning, it managed to clearly win after more feedback and instances were added. Also, the number of instances added at each iteration played a major role in changing the results. Adding an identical number of instances in each iteration of all four settings allowed the researcher to examine other factors that could affect the results and see whether/how they affected the results.

As mentioned earlier, active expert sourcing outperformed all the other settings after a number of iterations. This may suggest that having both expert sourcing and active learning (which is hypothesised as active expert sourcing) helps to increase the quality of the NER. Theoretically, in this case active learning led to the right decisions about what entities to choose, and including expert sourcing provided the correct class, which saved time and other resources (e.g. money, computation power, etc.)

Active crowdsourcing (+A−E)

Active learning with crowdsourcing (e.g. +A−E) and expert sourcing (e.g. −A+E) performed almost identically, and both had better performance than crowdsourcing alone (e.g. −A−E).

It can be assumed that the experts made excellent decisions since it resulted in performance that was as good as sampling with active learning. However, this may also suggest that the setting, *active crowdsourcing*, can also be considered as a good method for creating valuable resources, namely expert participation.

The concept of *active crowdsourcing* has already been used in other domains such as object recognition in images [Vijayanarasimhan and Grauman, 2014] and sentiment analysis [Brew et al., 2010]. It has also been applied to other natural language processing tasks such as NER [Laws et al., 2011].

7.3 Summary

In this chapter the author presents and explains the results of a real-world experiment in which a live online website was created and integrated into the GIS portal [Poesio et al., 2011b]. The four settings detailed in Chapter 6 were implemented to study the effects of active expert sourcing. The experiment involved running four live processing pipelines and comparing the outputs to each other. The measure used for evaluation was the F-score, as mentioned in Chapter 6.

Figure 7.10 shows that the proposed approach outperformed the other baselines. The key findings of this experiment are as follows:

- Portals for sharing publications can be approached as expert sourcing platforms on which experts who share their publications are asked to provide more information about or annotations of the work being shared.
- The effort needed to provide annotations can be reduced by utilising active learning; active learning is used to identify the most useful instances to be annotated by experts.
- The approach presented in this thesis, named active expert sourcing, outperforms other traditional methods in both data creation and knowledge extraction.

Table 7.10 Samples of annotated text in English.

The	10	-	CCHE	The	O	B-NP	O
animal	14	-	VI	animal	O	B-VX	O
bone	21	-	VF	bone	O	B-VX	O
industry	26	-	YF	industry	O	B-NP	O
coming	35	-	YF	coming	O	I-NP	O
from	42	-	YF	from	O	B-NP	O
Riparo	47	-	SPN	Riparo	B-PER	I-NP	B-Site
Cogola	54	-	SPN	Cogola	I-PER	B-NP	I-Site
-	61	-	XPO	-	O	O	O
The	63	-	VI	The	O	B-VX	O
author	67	-	VF	author	O	B-VX	O
presents	74	-	YF	presents	O	B-NP	O
the	83	-	YF	the	O	I-NP	O
animal	87	-	YF	animal	O	B-NP	O
-	93	-	XPO	-	O	I-NP	O
bone	94	-	SS	bone	O	I-NP	O
industry	99	-	AS	industry	O	I-NP	O
,	107	-	XPW	,	O	O	O
found	109	-	YF	found	O	B-NP	O
in	115	-	YF	in	O	I-NP	O
the	118	-	YF	the	O	B-NP	O
anthropic	122	-	YF	anthropic	O	I-NP	O
levels	132	-	YF	levels	O	B-NP	O
of	139	-	YF	of	O	I-NP	O
Riparo	142	-	SPN	Riparo	B-PER	B-NP	B-Site
Cogola	149	-	SPN	Cogola	I-PER	I-NP	I-Site
(	156	-	XPB	(	O	O	O
Carbonare	157	-	SPN	Carbonare	B-ORG	B-NP	B-Location
di	167	-	E	di	I-ORG	O	I-Location
Folgaria	170	-	SPN	Folgaria	I-ORG	B-NP	I-Location
,	178	-	XPW	,	O	O	O
Trento	180	-	SPN	Trento	B-GPE	B-NP	B-Location
)	186	-	XPB	)	O	O	O
during	188	-	YF	during	O	B-NP	O
the	195	-	YF	the	O	I-NP	O
1998	199	-	N	1998	O	I-NP	B-Absolutetime
-	203	-	XPO	-	O	I-NP	O
1999	204	-	N	1999	O	I-NP	O
field	209	-	YF	field	O	I-NP	O
researches	215	-	YF	researches	O	I-NP	O
.	225	<eos>	XPS	full_stop	O	B-NP	O

Table 7.11 Samples of annotated text in Italian.

RIASSUNTO	0	-	VSP	riassumere	O	B-VX	O
-	10	-	XPO	-	O	O	O
LOindustria	12	-	SPN	LÕindustria	O	B-NP	O
in	24	-	E	in	O	O	O
materia	27	-	SS	materia	O	B-NP	O
dura	35	-	VI	durare	O	B-VX	O
animale	40	-	AS	animale	O	I-VX	O
proveniente	48	-	B	proveniente	O	O	O
dal	60	-	ES	da/det	O	O	O
Riparo	64	-	SPN	Riparo	O	B-NP	B-Site
Cogola	71	-	SPN	Cogola	O	I-NP	I-Site
-	78	-	XPO	-	O	O	O
Si	80	-	PN	rifl	O	B-NP	O
presentano	83	-	VI	presentare	O	B-VX	O
di	94	-	E	di	O	O	O
seguito	97	-	SS	seguito	O	B-NP	O
i	105	-	RP	det	O	B-NP	O
manufatti	107	-	SP	manufatto	O	I-NP	B-Artefact
in	117	-	E	in	O	O	O
materia	120	-	SS	materia	O	B-NP	O
dura	128	-	VI	durare	O	B-VX	O
animale	133	-	B	animale	O	I-VX	O
rinvenuti	141	-	VPP	rinvenire	O	I-VX	O
nei	151	-	EP	in/det	O	O	O
livelli	155	-	SP	livello	O	B-NP	O
antropici	163	-	AP	antropico	O	I-NP	O
del	173	-	ES	di/det	O	O	O
Riparo	177	-	SPN	Riparo	O	B-NP	B-Site
Cogola	184	-	SPN	Cogola	O	I-NP	I-Site
(	191	-	XPB	(	O	O	O
Carbonare	192	-	SPN	Carbonare	O	B-NP	B-Location
di	202	-	E	di	O	O	O
Folgaria	205	-	SPN	Folgaria	B-GPE	B-NP	B-Location
,	213	-	XPW	,	O	O	O
Trento	215	-	SPN	Trento	B-GPE	B-NP	B-Location
)	221	-	XPB	)	O	O	O
durante	223	-	E	durante	O	O	O
le	231	-	RP	det	O	B-NP	O
campagne	234	-	SP	campagna	O	I-NP	O
di	243	-	E	di	O	O	O
scavo	246	-	SS	scavo	O	B-NP	O
1998	252	-	N	1998	O	I-NP	B-Absolutetime
-	256	-	XPO	-	O	I-NP	O
1999	257	-	N	1999	O	I-NP	B-Absolutetime
.	261	<eos>	XPS	full_stop	O	I-NP	O

Chapter 8

Active Expert Sourcing for Biochemistry

In the previous chapter (Chapter 7), the proposed *active expert sourcing* has been examined in a real-world experiment. An extensive implementation and evaluation shows that there is a potential for such an approach. This thesis proposes an approach that differs from existing methods, crowdsourcing and AL, in that is takes crowdsourcing a step back, and tries to implement crowdsourcing concepts in a different setting, whereby experts are the dominant worker/user types. Systems like Amazon Mechanical Turk (MTurk) and Crowdflower are known for their mostly non-expert workers. Platforms such as PRU GIS and ACL Anthology are known for their expert users. Being targeted in finding the right platform is a key component for *active expert sourcing*. Nevertheless, this thesis tries to reduce the resources needed for providing annotating by the use of AL.

Now that this proposed approach has showed potential in one domain (e.g. archaeology), this chapter tries to show to what extent the proposed approach can be generalised to other domains. This chapter is intended to examine *active expert sourcing*, on a larger scale. This is in line with the objectives discussed in Section 6.2.

Chapter structure

This chapter is structured as follows: Section 1 gives a general description of the experiment; Section 2 details the methods used to carry out the study, provides additional details about some of the parameter settings and the description of the data-set used, and Section 3 provides comprehensive details about the simulation utilised for this experiment; finally, Sections 4 and 5 present the results and the key findings of this experiment.

8.1 Introduction

Domains such as the biomedical and biochemical sciences require tools to help with the extraction of knowledge from text. The large amount of available text in such domains makes it very difficult to manually extract useful information. Therefore, automated methods are always preferable. Biomedical named entities, or chemical compounds and drugs, appear in collections of publications, such as research articles, patent reports, health service releases, or on the internet in general. PubMed, for instance, is one of the biggest collections that is freely accessible online. This makes it easy not only to search for publications and reports, but also to search for special components, such as chemical entities. Recognising this type of information, named entities, is a very crucial task in this particular domain. Identifying drug-drug interactions (DDI) and drug-protein interactions (DPI) depends heavily on the success of named-entity recognition.

In this chapter, the proposed method is applied on a larger scale, with the aim of gaining an enhanced understanding of how the proposed method performs in large-scale applications. However, this time, a different approach is taken, compared to experiments in the previous chapter. The experiments in this chapter use a simulating model rather than being live real-world experiments as in Chapter 7. This simulation approach has previously proven to be helpful when investigating an active learning implementation/algorithm [Sheng et al., 2008].

The availability of large data-sets, provided by tasks such as BioCreative [Krallinger et al., 2013] and BioNLP [Nédellec et al., 2013] has made it possible to carry out a large-scale implementation of our approach. Such data-set collections are labelled by domain experts, which, in our case, makes for a good fit.

8.2 Method

In the previous experiments, two sets of users were engaged: experts and non-experts; each user visited the portal and submitted one of their published works. These publications were processed by a pipeline to extract knowledge (e.g. NEs). Moreover, an AL algorithm was utilised to reduce the effort needed to increase performance, whereby, when the knowledge extraction pipeline is not confused, it asks the expert who uploaded the paper to give feedback on certain information/tokens. The aim of the experiments in this chapter was to simulate all of the above.

In this chapter's experiments, a simulation environment was built. As previously noted in the chapter concerning methodology (Chapter 6), there are number of simulation methods that are widely used in the literature to simulate annotators in crowdsourcing environments [Sheng et al., 2008][Raykar et al., 2009][Raykar et al., 2010]. Such simulation accounts for different user expertise when performing tasks [Yan et al., 2010b]. Therefore, a similar simulation model is used in this chapter.

The main idea for such a model is that simulated users provide a noisy version of the gold-standard answer. There are a number of studies that use this simulation model to study the issue where there is no gold-standard data; other studies examine the cost of requesting new labellings. In such a model, different levels of user expertise are demonstrated/simulated by adjusting the noise level. The model in [Yan et al., 2010b] is also similar to other key crowdsourcing simulations, such as [Sheng et al., 2008], [Raykar et al., 2009] and [Raykar et al., 2010].

In the procedure followed in this chapter, a collection of annotated text is obtained. This text is to be used as training, developing and testing sets. 10% of the data-set is an initial pool size; this is to kick start the AL module which is also used for the NER model. As users submitting text to a portal is simulated, it is simulated that every user would submit a portion that is equal to 10% of the data-set. This assumption is used in simulation similar to the one used in this chapter [Becker, 2008]. The NER itself is to be a CRF-based supervised learning algorithm. This algorithm has proven to be the state-of-the-art for sequence tagging. This algorithm is similar to the one used in Chapter 7 experiments.

Nevertheless, as a new domain is tackled, a new pipeline needs to be implemented. However, this thesis tries its best not to have to develop new software and tools from scratch, mainly because the intuition is that the proposed *active expert sourcing* is a general purpose approach that can be applied to different domain specific knowledge extraction scenarios with minimal effort. Basically, the thesis tackles the issue of when resources are short and access to experts is limited, therefore, it makes more sense to reduce any effort needed to a minimum. In the next sections, a detailed description of the implemented software/pipeline/system is provided and explained. However, before that, the simulation noise parameter is discussed below.

Nevertheless, as explained in Chapter 6, the F-score of each and every setting developed in the experiments is reported. Again, **-A -E** refers to Crowdsourcing, **+A -E** refers to Active Crowdsourcing, **+A +E** refers to Active Expert Sourcing and **-A +E** refers to Expert Sourcing.

8.2.1 Setting noise level

Many researchers have tried to address the issue of noise levels in crowdsourcing by using different methods. Some have simply set up a fixed noise level for their user modelling, while have others used some sort of kernel function to obtain a distribution of the noise level. For instance, [Yan et al., 2010b] and [Yan et al., 2012] assume that expert labelling coincides with the reality on the ground, and non-expert labelling results in having a 35% noise level. That assumption is relaxed later on in [Yan et al., 2011], where experts are set to have a level of noise at 10% and non-experts at 45%. On the other hand, a more sophisticated approach has also been applied, as in [Raykar et al., 2010], where a Gaussian noise model is used. However, in that study, two levels of noise were simulated: sensitivity and specificity. [Raykar et al., 2010] assumes that experts have between 10% and 20% noise and non-experts have between 40% and 45% noise. A different approach to introducing noise for multiple annotators is to draw it from a uniform distribution, as in [Kumar and Lease, 2011] and [Chen et al., 2013]. Moreover any noise can be introduced to affect only a specific part of the simulation, such as precision or recall [Vlachos, 2006].

Multiple studies addressing this parameter have made it clear that researchers are reluctant to accept any level below 50% noise. This might be due to the fact that, as [Angluin and Laird, 1988] in [Frénay and Verleysen, 2014] and [Sheng et al., 2008] suggest, with accuracy lower than 50%, no amount of training data can sufficiently increase performance in machine learning. At any rate, the way that noise is presented in data and the extent to which it actually affects machine learning algorithms [Nettleton et al., 2010] [Frénay and Verleysen, 2014] are issues beyond the scope of this study.

8.2.2 Implementation

In our environment, the assumption was that experts give labels that match gold-standard answers, while non-experts give labels that result in a noise level of 35%. This level of noise is similar to other AL simulation experiments in the literature [Yan et al., 2010b][Yan et al., 2012]. Later, the assumption about experts was modified, after they were found to label in a way that resulted in a noise level of 10% as in [Yan et al., 2011].

The data-set used has been released by BioCreative IV for the shared task CHEMDNER [Krallinger et al., 2013]. It contains a collection of publication abstracts in the field of biochemistry, collected from PubMed. To process the data, an active learning module is built and integrated into the system that scored best at the BioCreative IV CHEMDNER shared task [Lu et al., 2013]. The developed active learning module improves the learned model inside the processing pipeline at each iteration. This is done each time new data is added to a training set, as the active learning module queries annotators for feedback.

8.3 Description of the software developed

The basic idea is that a pipeline will take a text and process it, then an NER starts tagging. In the case that the NER is not certain about a token, it asks the expert. The whole process is done in a crowdsourcing like environment.

BioCreative IV for the shared task CHEMDNER [Krallinger et al., 2013] has received a number of submissions, each with a different NER implementation. The system that scored the top is available for download by the developers/authors [Lu et al., 2013].

In the developed pipeline for this chapter, there are different components. The majority of components come from the system that produced the best results at CHEMDNER shared task [Lu et al., 2013]. Firstly, the pipeline performs heavy pre-processing for the training data-set. Secondly, it learns two models using a CRF algorithm. One model is trained on the actual text, and the other is trained on an inverse letter-based segmentation of the text. When testing or tagging new text, the system pre-processes the text, tags it with the two models and post-processes it to produce a combined tab separated volume (`tsv`).

The same algorithm described and used in the previous experiments in Chapter 7 is used. In each iteration, the AL module selects the most informative instances, then asks for feedback from users.

The difference here from previous experiments is that an already annotated data-set is used to simulate the approach being presented. When the AL asks for feedback, the feedback is taken from the gold-standard data. The gold-standard data is annotated by experts; this simulates asking an expert for feedback, as in the previous real-world experiment.

However, before feedback is provided from the gold-standard data, some noise is added. This is according to the simulation model described earlier.

Now that a simulation environment has been set up, the experiment can be run. The initial NER which kick-starts the AL module, is trained on a subset of the data. Then the system simulates an expert submitting a document, which triggers the processing pipeline. The pipeline includes an AL algorithm that asks the simulated experts for feedback. The simulated experts provide feedback by getting the answer from the gold-standard data, however, with some noise level. Then this feedback gets added to the training set so that the AL NER model re-learns.

As there are four settings to be simulated, four pipelines are set up. These conform to the same settings discussed in Chapter 6 and in the experiments in Chapter 7. According to the baseline settings developed, there are two experiments that use crowdsourcing: one represent experts and the other, non-experts. These settings also include two types of NER: a random selection and AL NER selections . All run in a cycle, which is how AL is meant to work.

Nevertheless, at the end of each cycle, the NER is evaluated to check for improvements, compared to the baseline(s).

8.4 Results

Figure 8.1 shows the performance of all settings. It can be clearly seen that the proposed approach outperform other baselines. Also Table 8.1 shows a summary of the F-scores recorded at the experiment. The Friedman rank sum test was used to compute the $p - value$ for the 4 settings, and the settings started reporting a $p - value < 0.01$, from Iteration 2.

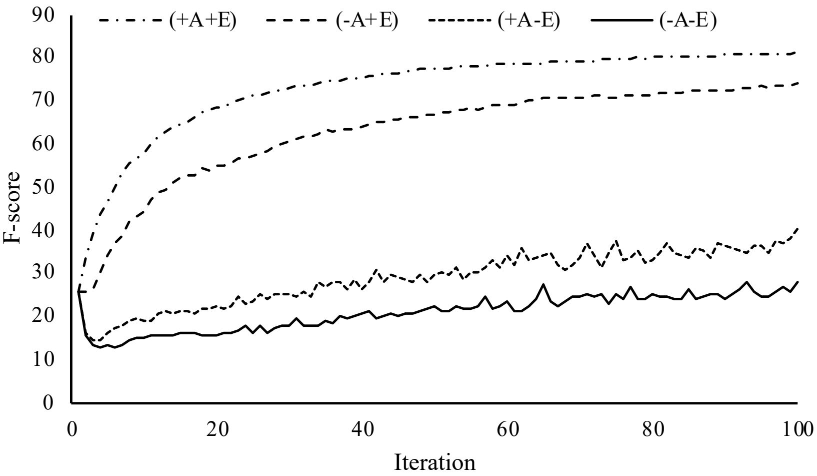


Fig. 8.1 F-scores for each setting, based on the number of instances added at each iteration.

The Wilcoxon signed rank test was computed to check for pair-wise significance. Starting from Iteration 2, there were significant differences between all settings. The Wilcoxon test for every setting against the others reported a $p - value < 0.01$.

Nevertheless, the experts' accuracy is relaxed, setting them at a 10% noise, per [Yan et al., 2010b] and [Yan et al., 2012]. The performance of the 4 settings is shown in Figure 8.2, which again reveals the extent to which the proposed approach outperformed others.

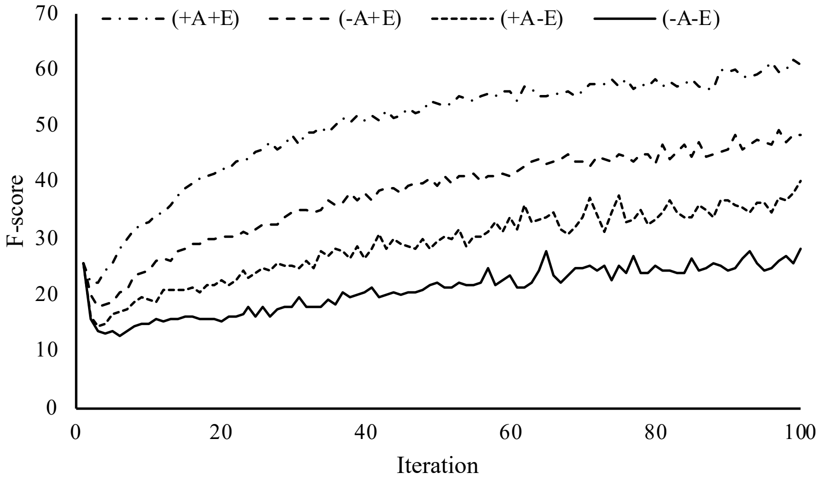


Fig. 8.2 The resulting F-scores after relaxing experts' accuracy where a 10% noise was added. Again, based on the number of instances added at each iteration.

Moreover, as with the first attempt, the results were found to be significant with the Friedman rank sum test only from the beginning of the experiment (e.g. Iteration 2). Further significant differences were found between all settings with the Wilcoxon signed rank test, from Iteration 2. The results of the Friedman test reported $p - value < 0.01$, and Wilcoxon test for every setting against the others also reported a $p - value < 0.01$.

8.5 Discussion

In contrast to the previous chapter, the discussion of results in this chapter is not divided into subsections, as the observations are discussed as a whole.

The results show that the proposed method, *active expert sourcing*, outperformed all other settings, though all settings managed to develop, and thereby reflect some improvement.

The first observation made possible by adapting *active expert sourcing*, is that experts managed to outperform non-experts –which is not surprising. However, this may, at least, indicate the appropriate application of the research methodology.

Since there was a control over the noise level, it was possible to demonstrate how performance heavily depends on the quality of the available data –thus on the expertise of annotators. This, perhaps, makes one question the real level of noise that both experts and non-experts demonstrate in the real world. In the literature chapters, some studies suggest that experts do have a certain level of expertise. The literature suggests that even experts can produce noise, due to, among other reasons, observing data that they are not familiar with. However, to this extent, this thesis is able to confidently claim that the proposed method can overcome such an issue. The model of applying active learning and crowdsourcing (*expert sourcing* in this case, e.g. −A+E), differs from the "traditional" application, and the differences can be summarised as follows:

- Expert source is happening in the right place/platforms, where the general expectation is that the majority of users are experts. The platforms proposed as a target are platforms such as ResearchGate and Academia. This is totally the opposite to using crowdsourcing platforms such as MTurk, where the majority of users/workers are non-expert.

- Experts are asked to provide the data that is to be added to the data-sets. Here, each expert provides a small pool of data that contributes to the total size of the data-set. Any noise that might be introduced by one expert *a)* is likely to be small compared to the overall size of the data-set, and *b)* might be overcome by the data provided by other experts.

- As each expert provides a small portion of the total data-set, reducing bias and increasing diversity can be achieved.

- Experts who produced the original text are asked to annotate their own text; this makes it highly unlikely for them to introduce noise due to the lack of domain knowledge/expertise.

It was also found that, from the results presented at Figure 8.2, each and every setting had an independent path; there were absolutely no intersections between any of the settings. It is assumed that this observation is derived from the fact that there was more control over these sets of experiments. This observation might be a direct result of fixing the parameters at each iteration. At each iteration, the same amount of instances are added to the training sets, and the same level of noise is presented. The reason behind this experiment decision is that, with the different settings being investigated, such parameters need to be locked to make for a fair comparison. It was noted from experiments in the archaeological domain (Chapter 7) that when these parameters change, so does performance tend to change.

However, having a fixed level of noise and a fixed number of instances might not represent the real world. Therefore these issues are regarded as a limitation of this sort of simulation and experimentation in this chapter.

Nonetheless, setting experts at no noise resulted in the two distinct pairs of lines in Figure 8.1, which meant a huge gap between experts and non-experts. The best of non-experts, a combination of active learning and crowdsourcing (e.g. +A−E), reached an F-score just above 0.40 after about 960 instances added.

After some noise is introduced to experts, at the level of 10%, the proposed approach still managed to outperform other approaches, as shown in Figure (8.2). However, this time, it was a tight competition; the difference between all settings was only about 0.1 in terms of the F-score. This might suggest that, in crowdsourcing, any enhancement, such as a utilising AL or involving experts, does make a difference in performance. Moreover, it can be seen that the noise level had a huge impact on the results. It can be seen that *active expert sourcing* dropped from above 0.80 to just about 0.60 after about 960 instances ware added.

8.6 Summary

In this chapter, a large-scale implementation of the hypothesised contribution of this thesis, *active expert sourcing*, is demonstrated. As in the previous chapter, four simultaneous pipelines ran, following the settings detailed in Chapter 6. All four settings managed to reflect some improvement, starting from a small pool of labelled examples. It has been shown that, unsurprisingly, experts always outperform non-experts. Moreover, the proposed approach did outperform all other baseline settings. The key findings in this experiment are summarised in the following points:

- *Active expert sourcing* can significantly outperform other settings presented in this thesis; this includes “traditional” active learning and crowdsourcing.
- Experts always outperform non-experts; this includes crowdsourcing compared to expert sourcing.
- *Active expert sourcing* can scale to different domains and perform significantly better.

- The claims on *active expert sourcing* performance in Chapter 7 stand, even in a larger scale set up and in a different domain.

Table 8.1 Summary of the F-scores of biochemistry experiment

Iteration	Instances	+A+E	-A+E	+A-E	-A-E
1	10	25.623	25.623	25.623	25.623
2	20	33.369	25.623	16.083	15.794
3	30	38.982	26.219	14.285	13.599
4	40	43.665	30.108	14.777	13.083
5	50	46.458	34.151	16.453	13.358
6	60	49.84	37.072	17.158	12.854
7	70	53.434	38.898	17.62	13.451
8	80	55.268	41.783	18.738	14.327
9	90	56.811	43.376	19.551	14.869
10	100	57.982	44.042	18.968	14.967
⋮	⋮	⋮	⋮	⋮	⋮
50	500	77.291	66.861	29.593	22.155
51	510	77.448	67.245	30.259	21.251
52	520	77.645	67.345	29.895	21.228
53	530	77.877	67.559	31.382	22.246
54	540	78.015	67.961	28.523	21.785
55	550	77.971	68.117	30.384	21.762
56	560	78.085	68.102	30.299	22.259
57	570	78.203	68.317	31.216	24.695
58	580	78.422	68.732	32.766	21.737
59	590	78.513	69.179	31.131	22.59
60	600	78.671	69.11	33.88	23.384
⋮	⋮	⋮	⋮	⋮	⋮
90	900	80.527	72.545	36.62	24.148
91	910	80.619	72.563	36.04	24.929
92	920	80.533	72.742	35.444	26.557
93	930	80.704	72.929	34.771	27.711
94	940	80.635	73.033	36.42	25.562
95	950	80.673	73.301	36.205	24.351
96	960	80.752	73.123	34.767	24.547
97	970	80.845	73.332	37.346	25.936
98	980	80.924	73.588	36.795	26.668
99	990	80.95	73.693	38.107	25.487
100	1000	81.124	73.866	40.342	28.031

Part IV

Concluding Remarks

Chapter 9

Conclusion and Future Work

Throughout this thesis, we have investigated the problem of resource creation and knowledge extraction from domain-specific information. Despite the fact that there has been some progress in addressing it, it is still a fundamental issue, not only because of the complexity of some domains, but because sciences are developing at a high speed. In this thesis, we have introduced a solution that we argue to have a potential. We have addressed the issue in the field of natural language processing, more specifically, named entity recognition. Within our proposed approaches, we have introduced two main contributions and addressed some novelties.

This thesis started with an introduction to the area of knowledge extraction from domain-specific information. We then introduced the elements that were to be utilised in our proposed approach. We argued that there are many platforms that can be easily used to recruit experts and can thus be used as expert sourcing platforms. We then introduced active learning as a way to reduce the effort needed from experts on such platforms to create resources.

After introducing the area of research, a comprehensive review of the key literature was presented and an explanation of how the current work fits into the research area was provided.

The approach introduced had a dedicated chapter (Chapter 6) that detailed the theoretical framework of this thesis. Chapter 6 also introduced the algorithm we developed specifically for the proposed approach – named active expert sourcing – in addition to describing the research methodology and experimental set up, and, lastly, outlining the testing and evaluation methods setting the ground for the evaluation framework.

The evaluation framework included two parts: a real-world experiment and a simulation experiment. Both experiments followed a well-established and conventional academic experimental set up.

In chapter 7, a real-world experiment was set up. The domain under examination was the digital humanities, specifically, the archaeological domain. The problems this domain faces, such as the lack of annotated and tagged resources, were identified, and an explanation of what makes such domains unique was provided. The chapter also highlighted the need for solutions that take such issues into account. For example, the type of entities that appear in the digital humanities differ from what appears in general text, which results in the fact that existing solutions are somehow difficult to be transported to such a domain, thus highlighting the need for domain-specific solutions. To tackle this issue, we developed a solution that integrates experts in a crowdsourcing platform to form an expert sourcing platform. We took advantage of the availability of platforms for sharing publications such as ResearchGate. It is known that such platforms attract many scientists, and many of them are happy to share their work there. This meant for us that such experts might also be happy to take part in providing more information about their own shared work. An experiment was run specifically to test this, and we were able to prove that this was feasible. We integrated an expert sourcing module into an existing platform for sharing publications in the archaeological domain. Our developed method also took advantage of active learning as a way to reduce the effort needed by experts to provide information. We tested our system in the real world and found that our approach had a significant impact on the process of creating high-quality resources. We managed to outperform different approaches such as traditional active learning and traditional crowdsourcing.

In Chapter 8, our approach was tested on a larger scale and in a different domain. The testing domain was the biochemistry domain. Again, and as mentioned earlier, we addressed the problem of named entity recognition. We used publicly available data-sets to run the experiments and we followed a well-established simulation approach to simulate crowdsourcing and expert sourcing. We then evaluated our results and found that we were again able to significantly outperform other approaches.

In the following sections, the contributions of this thesis are revisited, and suggestions for further research are stated. Lastly, the chapter is concluded with closing remarks.

9.1 Contributions of the thesis

- We identified a potential platform for expert sourcing.
- We applied active learning with expert sourcing.
- We introduced a novel active learning algorithm.
- We presented an extensive evaluation of the proposed approach.

9.2 Future work

The work presented in this thesis is only one step of a huge amount of research to be undertaken in the area of knowledge extraction for domain-specific information.

We tackled the issue of data creation and knowledge extraction for under-represented domains such as archaeology. However, there is always room to improve.

Although only one specific task was addressed in this thesis, which was named entity recognition, we believe there are many natural language processing tasks that can benefit from our proposed approach. Structure extraction, language detection, parts-of-speech tagging, and co-referencing, are some of the fields that can be easily adapted.

Natural language processing is also not the only field that can adapt our proposed approach. For instance, image tagging, face recognition and many other fields that address knowledge extraction can use the approach we introduced to accomplish any required task.

Lastly, established platforms for sharing publications among scholars can help in creating resourcing for all sorts of domains. The extra effort needed by experts to share their publications is not that significant, and we believe that one potential option is to work in collaboration with well-established platforms.

References

Aberdeen, J., Burger, J., Day, D., Hirschman, L., Robinson, P., and Vilain, M. (1995). Mitre: description of the alembic system used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 141–155. Association for Computational Linguistics.

Aker, A., El-Haj, M., Albakour, M.-D., Kruschwitz, U., et al. (2012). Assessing crowdsourcing quality through objective tasks. In *LREC*, pages 1456–1461. Citeseer.

Akkaya, C., Conrad, A., Wiebe, J., and Mihalcea, R. (2010). Amazon mechanical turk for subjectivity word sense disambiguation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 195–203. Association for Computational Linguistics.

Alagić, D. and Šnajder, J. (2015). Experiments on active learning for croatian word sense disambiguation. In *5th Workshop on Balto-Slavic Natural Language Processing associated with the 10th International Conference on Recent Advances in Natural Language Processing (RANLP 2015)*.

Alonso, O., Rose, D. E., and Stewart, B. (2008). Crowdsourcing for relevance evaluation. In *ACM SigIR Forum*, volume 42, pages 9–15. ACM.

Ambati, V. (2012). *Active Learning and Crowdsourcing for Machine Translation in Low Resource Scenarios*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA. AAI3528171.

Ambati, V., Vogel, S., and Carbonell, J. G. (2010). Active learning and crowd-sourcing for machine translation. In *LREC'10*. The Seventh International Conference on Language Resources and Evaluation.

Angluin, D. and Laird, P. (1988). Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Archak, N. (2010). Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on topcoder. com. In *Proceedings of the 19th international conference on World wide web*, pages 21–30. ACM.

Balasubramanian, V., Ho, S.-S., and Vovk, V. (2014). *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Newnes.

Baldwin, B., Collins, M., Eisner, J., Ratnaparkhi, A., Rosenzweig, J., and Sarkar, A. (1995). University of pennsylvania: description of the university of pennsylvania system used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 177–191. Association for Computational Linguistics.

Bartle, R. A. (2016). *MMOs from the Outside In*. Apress, 1 edition.

Becker, M. (2008). *Active learning an explicit treatment of unreliable parameters*. Thesis or dissertation, Institute for Communicating and Collaborative Systems.

Bloodgood, M. and Callison-Burch, C. (2010). Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 854–864. Association for Computational Linguistics.

Bloodgood, M. and Grothendieck, J. (2015). Analysis of stopping active learning based on stabilizing predictions. *arXiv preprint arXiv:1504.06329*.

Bonin, F., Cavulli, F., Noriller, A., Poesio, M., and Stemle, E. W. (2012). Annotating archaeological texts: an example of domain-specific annotation in the humanities. In *Proceedings of the Sixth Linguistic Annotation Workshop*, LAW VI '12, pages 134–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bouguelia, M.-R., Belaïd, Y., and Belaïd, A. (2016). An adaptive streaming active learning strategy based on instance weighting. *Pattern Recognition Letters*, 70:38–44.

Brew, A., Greene, D., and Cunningham, P. (2010). Using crowdsourcing and active learning to track sentiment in online media. In *ECAI*, pages 145–150.

Bunescu, R., Ge, R., Mooney, R. J., Marcotte, E., and Ramani, A. K. (2002). Extracting gene and protein names from biomedical abstracts. *Unpublished Technical Note, Available from http://www. cs. utexas. edu/users/ml/publication/ie. html*.

Bunescu, R. C. and Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *Eacl*, volume 6, pages 9–16.

Cai, W., Zhang, M., and Zhang, Y. (2015). Active learning for ranking with sample density. *Information Retrieval Journal*, 18(2):123–144.

Cai, W., Zhang, Y., Zhou, S., Wang, W., Ding, C., and Gu, X. (2014). Active learning for support vector machines with maximum model change. In *Machine Learning and Knowledge Discovery in Databases*, pages 211–226. Springer.

Callison-Burch, C. and Dredze, M. (2010). Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12. Association for Computational Linguistics.

Cardellino, C., Alemany, L. A., Villata, S., and Cabrio, E. (2015a). Improvements in information extraction in legal text by active learning. In *Proceedings of the 28th Annual Conference on Legal Knowledge and Information Systems*, pages 21–30.

Cardellino, C., Villata, S., Alemany, L. A., and Cabrio, E. (2015b). Information extraction with active learning: A case study in legal text. In *Computational Linguistics and Intelligent Text Processing*, pages 483–494. Springer.

Carpenter, B. (2007). Lingpipe for 99.99% recall of gene mentions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 307–309.

Carterette, B. and Soboroff, I. (2010). The effect of assessor error on ir system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 539–546. ACM.

Chamberlain, J., Poesio, M., and Kruschwitz, U. (2008). Phrase detectives: A web-based collaborative annotation game. In *Proceedings of the International Conference on Semantic Systems (I-Semantics' 08)*, pages 42–49.

Chan, Y. S. and Ng, H. T. (2007). Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, volume 45, pages 49–56.

Chen, J., Schein, A., Ungar, L., and Palmer, M. (2006). An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 120–127. Association for Computational Linguistics.

Chen, X., Bennett, P. N., Collins-Thompson, K., and Horvitz, E. (2013). Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202. ACM.

Chen, Y., Lasko, T. A., Mei, Q., Denny, J. C., and Xu, H. (2015). A study of active learning methods for named entity recognition in clinical text. *Journal of biomedical informatics*, 58:11–18.

Cohen, W. W. and Sarawagi, S. (2004). Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine learning*, 15(2):201–221.

Cohn, D. A. (1994). *Neural network exploration using optimal experiment design.* Artificial Intelligence Lab - MIT.

Costa, J., Silva, C., Antunes, M., and Ribeiro, B. (2011). On using crowdsourcing and active learning to improve classification performance. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pages 469–474. IEEE.

Cowie, J. (1995). Crl/nmsu: description of the crl/nmsu systems used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 157–166. Association for Computational Linguistics.

Cucerzan, S. and Yarowsky, D. (1999). Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, pages 90–99.

Dagan, I. and Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. The Morgan Kaufmann series in machine learning,(San Francisco, CA, USA).

Demartini, G., Difallah, D. E., and Cudré-Mauroux, P. (2012). Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478. ACM.

Di, W. and Crawford, M. M. (2012). View generation for multiview maximum disagreement based active learning for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(5):1942–1954.

Dictionary, L. (2016). Longman dictionary of contemporary english.

Dini, M. and Sagramoni, A. (2005). Analisi dei prodotti della scheggiatura del sito dell'epigravettiano finale di la greppia ii–us 1 (parco naturale dell'orecchiella–lucca). *Preistoria Alpina*, 41:5–21.

DiPalantino, D. and Vojnovic, M. (2009). Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 119–128. ACM.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S., and Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*, volume 2, pages 837–840.

Downs, J. S., Holbrook, M. B., Sheng, S., and Cranor, L. F. (2010). Are your participants gaming the system?: screening mechanical turk workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2399–2402. ACM.

Du, J., Wang, M., and Zhang, M. (2014). Sentence-length informed method for active learning based resource-poor statistical machine translation. In *Natural Language Processing and Chinese Computing*, pages 91–102. Springer.

Du, J., Zhang, Z., Yan, J., Cui, Y., and Chen, Z. (2010). Using search session context for named entity recognition in query. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 765–766. ACM.

Eickhoff, C. and de Vries, A. (2011). How crowdsourcable is your task. In *Proceedings of the workshop on crowdsourcing for search and data mining (CSDM) at the fourth ACM international conference on web search and data mining (WSDM)*, pages 11–14.

Ekbal, A., Bonin, F., Saha, S., Stemle, E., Barbu, E., Cavulli, F., Girardi, C., and Poesio, M. (2011). Rapid adaptation of ne resolvers for humanities domains using active annotation. *JLCL*, 26(2):29–51.

Ekbal, A., Bonin, F., Saha, S., Stemle, E., Barbu, E., Cavulli, F., Girardi, C., and Poesio, M. (2012a). Rapid adaptation of ne resolvers for humanities domains using active annotation. *JOURNAL FOR LANGUAGE TECHNOLOGY AND COMPUTATIONAL LINGUISTICS*, 26(2):1–12.

Ekbal, A., Saha, S., and Singh, D. (2012b). Active machine learning technique for named entity recognition. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pages 180–186. ACM.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88. Association for Computational Linguistics.

Finn, A. and Kushmerick, N. (2003). Active learning selection strategies for information extraction. In *Proceedings of the International Workshop on Adaptive Text Extraction and Mining (ATEM-03)*, pages 18–25.

Fleischman, M. (2001). Automated subcategorization of named entities. In *ACL (Companion Volume)*, pages 25–30.

Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: a survey. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(5):845–869.

Fu, Y., Zhu, X., and Li, B. (2013). A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283.

Gangadharaiah, R., Brown, R. D., and Carbonell, J. (2009). Active learning in example-based machine translation. In *Proceedings of the 17th Nordic Conference of Computational Linguistics NODALIDA*, volume 227, page 30.

Gao, Q. and Vogel, S. (2010). Consensus versus expertise: A case study of word alignment with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 30–34. Association for Computational Linguistics.

Gasperin, C. (2009). Active learning for anaphora resolution. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 1–8. Association for Computational Linguistics.

Gillick, D. and Liu, Y. (2010). Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151. Association for Computational Linguistics.

Gordon, J., Van Durme, B., and Schubert, L. K. (2010). Evaluation of commonsense knowledge with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 159–162. Association for Computational Linguistics.

Grady, C. and Lease, M. (2010). Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk*, pages 172–179. Association for Computational Linguistics.

Gramates, L. S., Marygold, S. J., dos Santos, G., Urbano, J.-M., Antonazzo, G., Matthews, B. B., Rey, A. J., Tabone, C. J., Crosby, M. A., Emmert, D. B., et al. (2016). Flybase at 25: looking to the future. *Nucleic Acids Research*, page gkw1016.

Grishman, R. (1995). The nyu system for muc-6 or where's the syntax? In *Proceedings of the 6th conference on Message understanding*, pages 167–175. Association for Computational Linguistics.

Grishman, R. and Sundheim, B. (1996a). Design of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, pages 413–422. Association for Computational Linguistics.

Grishman, R. and Sundheim, B. (1996b). Message understanding conference-6: A brief history. In *Coling*, volume 96, pages 466–471.

Grönroos, S.-A., Jokinen, K., Hiovain, K., Kurimo, M., and Virpioja, S. (2015). Low-resource active learning of north sámi morphological segmentation. In *Septentrio Conference Series*, volume 2, pages 20–33.

Guo, J., Xu, G., Cheng, X., and Li, H. (2009). Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM.

Hady, M. F. A., Karali, A., Kamal, E., and Ibrahim, R. (2014). Unsupervised active learning of crf model for cross-lingual information extraction. *International Journal of Computational Linguistics and Applications*, 5(2):95.

Haffari, G., Roy, M., and Sarkar, A. (2009). Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 415–423. Association for Computational Linguistics.

Hajmohammadi, M. S., Ibrahim, R., Selamat, A., and Fujita, H. (2015). Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples. *Information Sciences*, 317:67–77.

Hakkani-Tur, D., Riccardi, G., and Gorin, A. (2002). Active learning for automatic speech recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3904. IEEE.

Han, K.-S., Chung, H.-J., Kim, S.-B., Song, Y.-I., Lee, J.-Y., and Rim, H.-C. (2004). Korea university question answering system at trec 2004. In *TREC*.

Hanneke, S. (2014). Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309.

Harris, C. (2011). You're hired! an examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 15–18.

Heer, J. and Bostock, M. (2010). Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 203–212. ACM.

Hirschman, L. (1991). Comparing muck-ii and muc-3: Assessing the difficulty of different tasks. In *Proceedings of the 3rd conference on Message understanding*, pages 25–30. Association for Computational Linguistics.

Hirth, M., Hoßfeld, T., and Tran-Gia, P. (2010). Cheat-detection mechanisms for crowdsourcing. *University of Würzburg, Tech. Rep*, 4.

Horton, J. J. and Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 209–218. ACM.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

Huang, E., Zhang, H., Parkes, D. C., Gajos, K. Z., and Chen, Y. (2010). Toward automatic task design: a progress report. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 77–85. ACM.

Huang, F. J. and LeCun, Y. (2006). Large-scale learning with svm and convolutional for generic object categorization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 284–291.

Huberman, B. A., Romero, D. M., and Wu, F. (2009). Crowdsourcing, attention and productivity. *Journal of Information Science.*

Ipeirotis, P. G. (2010a). Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21.

Ipeirotis, P. G. (2010b). Demographics of mechanical turk. *NYU Working Paper*, (CEDER-10-01).

Ipeirotis, P. G., Provost, F., and Wang, J. (2010). Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM.

Iwafiska, L. (1992). A general semantic model of negation in natural language: Representation and inference. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR92)*, pages 357–368.

Iwańska, L. (1993). Logical reasoning in natural language: It is all about knowledge. *Minds and Machines*, 3(4):475–510.

Iwańska, L. (1995). Wayne state university: Description of the uno natural language processing system as used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 263–277. Association for Computational Linguistics.

Iwarlska, L. (1994). Talking about time: Temporal reasoning as a problem of natural language. Technical Report FS-94-04, AAAI Technical Report.

Jagadeesan, A. P., Lynn, A., Corney, J. R., Yan, X., Wenzel, J., Sherlock, A., and Regli, W. (2009). Geometric reasoning via internet crowdsourcing. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 313–318. ACM.

Jain, S. and Parkes, D. C. (2009). The role of game theory in human computation systems. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 58–61. ACM.

Jha, M., Andreas, J., Thadani, K., Rosenthal, S., and McKeown, K. (2010). Corpus creation for new genres: A crowdsourced approach to pp attachment. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 13–20. Association for Computational Linguistics.

Ji, H., Nothman, J., Dang, H. T., and Hub, S. I. (2016). Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*.

Ji, H., Nothman, J., Hachey, B., and Florian, R. (2015). Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

Karger, D. R., Oh, S., and Shah, D. (2014). Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24.

Kazai, G. (2010). An exploration of the influence that task parameters have on the performance of crowds. *Proceedings of the CrowdConf*, 2010.

Kholghi, M., Sitbon, L., Zuccon, G., and Nguyen, A. (2015). External knowledge and query strategies in active learning: a study in clinical information extraction. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 143–152. ACM.

Kim, H.-G., Roh, J., Lee, H., Kim, G., and Lee, S.-Y. (2015). Active learning for large-scale object classification: from exploration to exploitation. In *Proceedings of the 3rd International Conference on Human-Agent Interaction*, pages 251–254. ACM.

Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004). Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Association for Computational Linguistics.

Kim, S., Song, Y., Kim, K., Cha, J.-W., and Lee, G. G. (2006). Mmr-based active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 69–72. Association for Computational Linguistics.

Kittur, A., Chi, E. H., and Suh, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM.

Koblin, A. M. (2009). The sheep market. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 451–452. ACM.

Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., and Valencia, A. (2013). Overview of the chemical compound and drug name recognition (chemdner) task. In *BioCreative Challenge Evaluation Workshop*, volume 2, page 2.

Kranjc, J., Smailović, J., Podpečan, V., Grčar, M., Žnidaršič, M., and Lavrač, N. (2015). Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, 51(2):187–203.

Kripke, S. A. (1980). *Naming and necessity.* Harvard University Press.

Kumar, A. and Lease, M. (2011). Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 19–22.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Laws, F., Heimerl, F., and Schütze, H. (2012). Active learning for coreference resolution. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 508–512. Association for Computational Linguistics.

Laws, F., Scheible, C., and Schütze, H. (2011). Active learning with amazon mechanical turk. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1546–1556, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leaman, R., Gonzalez, G., et al. (2008). Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific symposium on biocomputing*, volume 13, pages 652–663. Big Island, Hawaii.

Lease, M. (2011). On quality control and machine learning in crowdsourcing. *Human Computation*, 11:11.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, C., Hwang, Y.-G., and Jang, M.-G. (2007). Fine-grained named entity recognition and relation extraction for question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 799–800. ACM.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12, New York, NY, USA. Springer-Verlag New York, Inc.

Li, J., Bioucas-Dias, J. M., and Plaza, A. (2013a). Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(2):844–856.

Li, S., Ju, S., Zhou, G., and Li, X. (2012). Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148. Association for Computational Linguistics.

Li, S., Xue, Y., Wang, Z., and Zhou, G. (2013b). Active learning for cross-domain sentiment classification. In *IJCAI*.

Li, X. and Guo, Y. (2013). Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866.

Lin, D. (1993). University of manitoba: Description of the nuba system as used for muc-5. In *Proceedings of the 5th conference on Message understanding*, pages 263–275. Association for Computational Linguistics.

Lin, D. (1995). University of manitoba: description of the pie system used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 113–126. Association for Computational Linguistics.

Ling, X. and Weld, D. S. (2012). Fine-grained entity recognition. In *AAAI*.

Logacheva, V. and Specia, L. (2014). Confidence-based active learning methods for machine translation. *EACL 2014*, page 78.

Lu, Y., Yao, X., Wei, X., and Ji, D. (2013). Whu-bionlp chemdner system with mixed conditional random fields and word clustering. In *Proceedings of the Fourth BioCreative Challenge Evaluation Workshop*, volume 2, pages 129–134.

Ma, Y., Garnett, R., and Schneider, J. (2013). σ-optimality for active learning on gaussian random fields. In *Advances in Neural Information Processing Systems*, pages 2751–2759.

Magerman, D. M. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 276–283. Association for Computational Linguistics.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Mason, W. and Watts, D. J. (2010). Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., and Wilks, Y. (2001). Named entity recognition from diverse text types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274.

McDonald, D. (1996). Internal and external evidence in the identification and semantic categorization of proper names. *Corpus processing for lexical acquisition*, pages 21–39.

Mellebeek, B., Benavent, F., Grivolla, J., Codina, J., Costa-Jussa, M. R., and Banchs, R. (2010). Opinion mining of spanish customer comments with non-expert annotations on mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on Creating speech and language data with Amazon's mechanical turk*, pages 114–121. Association for Computational Linguistics.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Miller, T. A., Dligach, D., and Savova, G. K. (2012). Active learning for coreference resolution. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 73–81. Association for Computational Linguistics.

Moreno, A., de la Rosa, J. L., Szymanski, B. K., and Barcenas, J. M. (2009). Reward system for completing faqs. In *CCIA*, pages 361–370.

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

Morgan, R., Garigliano, R., Callaghan, P., Poria, S., Smith, M., and Cooper, C. (1995). University of durham: Description of the lolita system as used in muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 71–85. Association for Computational Linguistics.

Mori, S., Suen, C. Y., and Yamamoto, K. (1992). Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7):1029–1058.

Moskovitch, R., Nissim, N., Stopel, D., Feher, C., Englert, R., and Elovici, Y. (2007). Improving the detection of unknown computer worms activity using active learning. In *KI 2007: Advances in Artificial Intelligence*, pages 489–493. Springer.

Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., and Faloutsos, C. (2013). Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 886–893. IEEE.

Nam, K. K., Ackerman, M. S., and Adamic, L. A. (2009). Questions in, knowledge in?: a study of naver's question answering community. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 779–788. ACM.

Nédellec, C., Bossy, R., Kim, J.-D., Kim, J.-J., Ohta, T., Pyysalo, S., and Zweigenbaum, P. (2013). Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7.

Nepil, M., Popelínsky, L., Popel, L., et al. (2001). Part-of-speech tagging by means of ilp and active learning. In *Proceedings of the Workshop on Instance Selection at ECML/PKDD*. Department of Computer Science, Albert-Ludwigs University, Citeseer.

Nettleton, D. F., Orriols-Puig, A., and Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review*, 33(4):275–306.

Parent, G. and Eskenazi, M. (2010). Clustering dictionary definitions using amazon mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 21–29. Association for Computational Linguistics.

Paşca, M. (2007). Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 683–690. ACM.

Pedrycz, W., Koronacki, J., Raś, Z. W., Wierzchoń, S. T., and Kacprzyk, J. (2010). *Advances in Machine Learning II: Dedicated to the Memory of Professor Ryszard S. Michalski*. Springer-Verlag Berlin Heidelberg.

Piskorski, J., Pivovarova, L., Šnajder, J., Steinberger, J., and Yangarber, R. (2017). The first cross-lingual challenge on recognition, normalization and matching of named entities in slavic languages. *BSNLP 2017*, page 76.

Poesio, M., Barbu, E., Bonin, F., Cavulli, F., Ekbal, A., Girardi, C., Nardelli, F., Saha, S., and Stemle, E. (2011a). The humanities research portal: Human language technology meets humanities publication repositories. *Proceedings of Supporting Digital Humanitites (SDH), Copenhagen.*

Poesio, M., Barbu, E., Bonin, F., Cavulli, F., Ekbal, A., Girardi, C., Nardelli, F., Saha, S., and Stemle, E. (2011b). The humanities research portal: Human language technology meets humanities publication repositories. *Proceedings of Supporting Digital Humanitites (SDH), Copenhagen.*

Poesio, M., Barbu, E., Stemle, E., and Girardi, C. (2011c). Structure-preserving pipelines for digital libraries. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 54–62. Association for Computational Linguistics.

Qi, Y. and Zhang, G. (2016). Strategy of active learning support vector machine for image retrieval. *Computer Vision, IET*, 10(1):87–94.

Quinlan, J. (1991). Machine learning: Easily understood decision rules. *Computer Systems that Learn, eds. Weiss, SM and Kulikowski, CA, Morgan Kaufmann.*

Quinlan, J. R. (1979). Discovering rules by induction from large collections of example. *Expert Systems in the Micro Electronics Age.*

Rau, L. (1994). Method for extracting company names from text. US Patent 5,287,278.

Rau, L. F. (1991). Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume 1, pages 29–32. IEEE.

Raykar, V. C., Yu, S., Zhao, L. H., Jerebko, A., Florin, C., Valadez, G. H., Bogoni, L., and Moy, L. (2009). Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th annual international conference on machine learning*, pages 889–896. ACM.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322.

Reker, D. and Schneider, G. (2015). Active-learning strategies in computer-assisted drug discovery. *Drug discovery today*, 20(4):458–465.

Riccardi, G. and Hakkani-Tür, D. (2005). Active learning: Theory and applications to automatic speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 13(4):504–511.

Rijsbergen, C. J. V. (1979). *Information Retrieval.* Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

Ringger, E., McClanahan, P., Haertel, R., Busby, G., Carmen, M., Carroll, J., Seppi, K., and Lonsdale, D. (2007a). Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 101–108. Association for Computational Linguistics.

Ringger, E., McClanahan, P., Haertel, R., Busby, G., Carmen, M., Carroll, J., Seppi, K., and Lonsdale, D. (2007b). Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, LAW '07, pages 101–108, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robertson, M. (2010). Can't play, won't play. In *http://hideandseek.net/2010/10/06/cant-play-wont-play/.*

Ross, J., Irani, L., Silberman, M., Zaldivar, A., and Tomlinson, B. (2010). Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 extended abstracts on Human factors in computing systems*, pages 2863–2872. ACM.

Roy, N. and McCallum, A. (2001). Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448.

Sachan, M., Hovy, E., and Xing, E. P. (2015). An active learning approach to coreference resolution. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 1312–1318.

Settles, B. (2005). Abner: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY, USA. ACM.

Shannon, C. E. and Weaver, W. (2015). *The mathematical theory of communication*. University of Illinois press.

Sheng, V. S., Provost, F., and Ipeirotis, P. G. (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM.

Silberman, M., Irani, L., and Ross, J. (2010). Ethics and tactics of professional crowdwork. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):39–43.

Singh, V. K., Jain, R., and Kankanhalli, M. S. (2009). Motivating contributors in social media networks. In *Proceedings of the first SIGMM workshop on Social media*, pages 11–18. ACM.

Smailović, J., Grčar, M., Lavrač, N., and Žnidaršič, M. (2014). Stream-based active learning for sentiment analysis in the financial domain. *Information Sciences*, 285:181–203.

Smatana, M., Koncz, P., Smatana, P., and Paralic, J. (2013). Active learning enhanced semi-automatic annotation tool for aspect-based sentiment analysis. In *Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on*, pages 191–194. IEEE.

Sundheim, B. M. (1996). Overview of results of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, pages 423–442. Association for Computational Linguistics.

Tanev, H. and Magnini, B. (2006). Weakly supervised approaches for ontology population. In *11th Conference of the European Chapter of the Association for Computational Linguistics.*

Tang, W. and Lease, M. (2011). Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR).*

Thompson, C. A., Califf, M. E., and Mooney, R. J. (1999). Active learning for natural language parsing and information extraction. In *ICML*, pages 406–414. Citeseer.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Vezhnevets, A., Buhmann, J. M., and Ferrari, V. (2012). Active learning for semantic segmentation with expected change. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3162–3169. IEEE.

Vijayanarasimhan, S. and Grauman, K. (2014). Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114.

Vinayakarao, V., Sarma, A., Purandare, R., Jain, S., and Jain, S. (2017). Anne: Improving source code search using entity retrieval approach. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 211–220. ACM.

Vlachos, A. (2004). Active learning with support vector machines. *Master of Science School of Informatics University of Edinburgh.*

Vlachos, A. (2006). Active annotation. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 64–71.

Vlachos, A. (2007a). Evaluating and combining biomedical named entity recognition systems. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 199–206. Association for Computational Linguistics.

Vlachos, A. (2007b). Tackling the biocreative2 gene mention task with conditional random fields and syntactic parsing. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop; 23 to 25 April 2007; Madrid, Spain*, pages 85–87.

Vlachos, A. (2008). A stopping criterion for active learning. *Computer Speech & Language*, 22(3):295–312.

Vlachos, A. and Gasperin, C. (2006). Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 138–145. Association for Computational Linguistics.

Vlachos, A., Gasperin, C., Lewin, I., and Briscoe, T. (2006). Bootstrapping the recognition and anaphoric linking of named entities in drosophila articles. In *Biocomputing 2006*, pages 100–111. World Scientific.

Voorhees, E. M. et al. (1999). The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

Voyer, R., Nygaard, V., Fitzgerald, W., and Copperman, H. (2010). A hybrid model for annotating named entity training corpora. In *Proceedings of the fourth linguistic annotation workshop*, pages 243–246. Association for Computational Linguistics.

Wang, R. and Callison-Burch, C. (2010). Cheap facts and counter-facts. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 163–167. Association for Computational Linguistics.

Weischedel, R. (1995). Ben: description of the plum system as used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 55–69. Association for Computational Linguistics.

Weischedel, R. and Brunstein, A. (2005). Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112.

Weisstein, E. W. (2004). Bonferroni correction.

Welinder, P. and Perona, P. (2010). Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE.

Woolley, J., Madsen, T. L., and Sarangee, K. (2015). Crowdsourcing or expertsourcing: Building and engaging online communities for innovation? In *DRUID15*.

Wu, F., Wilkinson, D. M., and Huberman, B. A. (2009). Feedback loops of attention in peer production. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 409–415. IEEE.

Yan, T., Kumar, V., and Ganesan, D. (2010a). Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 77–90. ACM.

Yan, Y., Fung, G. M., Rosales, R., and Dy, J. G. (2011). Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168.

Yan, Y., Rosales, R., Fung, G., and Dy, J. (2012). Modeling multiple annotator expertise in the semi-supervised learning scenario. *arXiv preprint arXiv:1203.3529*.

Yan, Y., Rosales, R., Fung, G., Schmidt, M. W., Valadez, G. H., Bogoni, L., Moy, L., and Dy, J. G. (2010b). Modeling annotator expertise: Learning when everybody knows a bit of something. In *International conference on artificial intelligence and statistics*, pages 932–939.

Yang, J., Adamic, L. A., and Ackerman, M. S. (2008). Crowdsourcing and knowledge sharing: strategic user behavior on taskcn. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 246–255. ACM.

Yang, Y., Ma, Z., Nie, F., Chang, X., and Hauptmann, A. G. (2015). Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127.

You, X., Wang, R., and Tao, D. (2014). Diverse expected gradient active learning for relative attributes. *Image Processing, IEEE Transactions on*, 23(7):3203–3217.

Yu, D., Varadarajan, B., Deng, L., and Acero, A. (2010). Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Computer Speech & Language*, 24(3):433–444.

Yuen, M.-C., Chen, L.-J., and King, I. (2009). A survey of human computation systems. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 723–728. IEEE.

Yuen, M.-C., King, I., and Leung, K.-S. (2011). Task matching in crowdsourcing. In *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 409–412. IEEE.

Zhai, K., Kozareva, Z., Hu, Y., Li, Q., and Guo, W. (2016). Query to knowledge: Unsupervised entity extraction from shopping queries using adaptor grammars. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 255–264. ACM.

Zhang, B., Wang, Y., and Chen, F. (2014). Multilabel image classification via high-order label correlation driven active learning. *Image Processing, IEEE Transactions on*, 23(3):1430–1441.

Zhang, H.-T., Huang, M.-L., and Zhu, X.-Y. (2012). A unified active learning framework for biomedical relation extraction. *Journal of Computer Science and Technology*, 27(6):1302–1313.

Zhang, J. J., Chan, R. H. Y., and Fung, P. (2009). Extractive speech summarization by active learning. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 392–397. IEEE.

Zhang, J. J. and Fung, P. (2012). Active learning with semi-automatic annotation for extractive speech summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 8(4):6.

Zhang, Y., Yang, H. L., Prasad, S., Pasolli, E., Jung, J., and Crawford, M. (2015). Ensemble multiple kernel active learning for classification of multisource remote sensing data. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 8(2):845–858.

Zhao, S. and Ng, H. T. (2014). Domain adaptation with active learning for coreference resolution. *EACL 2014*, pages 21–29.

Zhao, Z. and Ma, X. (2013). Active learning for speech emotion recognition using conditional random fields. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on*, pages 127–131. IEEE.

Zheng, H., Li, D., and Hou, W. (2011). Task design, motivation, and participation in crowdsourcing contests. *International Journal of Electronic Commerce*, 15(4):57–88.

Zhu, J. and Hovy, E. H. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, volume 7, pages 783–790.

Zhu, J., Wang, H., Yao, T., and Tsou, B. K. (2008). Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1137–1144. Association for Computational Linguistics.

Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.