Title (EN): Domain-Specific NER Adaptation

In the past years, the Named Entity Recognition (NER) technology has been under an active development and enjoy a significant increase in popularity and usage in the academic and industrial sphere. Nevertheless, vast majority of the developed NER systems have been developed as general-purpose systems. While they can perform well on multiple domains (macro level), on specific domains (micro level) their performance quality might be low. The ultimate goal of the thesis is to develop domain-specific NER models. Guidelines:

- Get familiar with the NER technology and available NER frameworks.
- Investigate possible datasets for domain-specific training of NER.
- Develop NER training datasets for several selected domains (e.g. sports, politics, music, etc.).
- Train a domain-specific NER model using existing frameworks, such as DBpedia Spotlight or StanfordNER.
- Validate and evaluate the developed domain-specific NER models.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Domain-specific Named Entity Recognition

Bc. Bogoljub Jakovcheski

Department of software engineering
Supervisor: Ing. Milan Dojchinovski, Ph.D.

May 29, 2018

Acknowledgements

I would like to thank my family and friends for support during writing this thesis.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on May 29, 2018 .....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Jakovcheski, Bogoljub. *Domain-specific Named Entity Recognition.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2018.

Abstrakt

Klíčová slova

Abstract

Keywords

Contents

Citation of this thesis vi

Introduction **1**
Motivation 1
Goals of the thesis 1
Thesis outline 1

1 Background and related work **3**
1.1 Information extraction 3
1.2 Named Entity Recognition 4
1.2.1 F_1 score 5
1.2.2 Stanford NER 5
1.3 RDF/NIF 6
1.4 Domain specific Named Entity Recognition 6
1.5 DBpedia ontology 7
1.6 Evaluation/training datasets 7
1.7 HDT 8
1.8 Apache Jena 8

2 Domain specific named entity recognition **9**
2.1 Data pre-processing 9
2.2 Domain specification 11
2.3 Types retrieval 11
2.4 Data transformation/output for Stanford 14

3 Experiments **15**
3.1 Goals of the experiments 15
3.2 List of experiments 15
3.2.1 Main experiment 15
3.2.2 Experiments that have lower number of traning abstracts in model 21

3.2.3 Experiments that have lower number of traning abstracts in model 36
3.2.4 MIXED 45
3.2.5 Experiments with lower training abstracts on model, but higher abstracts on test file 48
3.2.6 Experiments with higher training abstracts on model, but lower abstracts on test file 56
3.3 Summary of results 67
3.3.1 Graphs 67

Conclusion **69**

Bibliography **71**

A Contents of CD **73**

List of Figures

1.1 Information extraction, downloaded from https://www.slideshare.net/rubenizquierdobevia/inform extraction-45392844 4
1.2 Stanford NER GUI with 3 classes model (Location, Person, Organization) 5
1.3 Dbpedia Ontology - Instances per class 7

3.1 All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 100 Links 48
3.2 All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 300 Links 49
3.3 All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 500 Links 49
3.4 All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 500 Links With Lower PageRank . . . 50
3.5 All 3 Domains Fine Grained Top 100 Links Runned With All 3 Domains Fine Grained Top 500 Links 50
3.6 All 3 Domains Fine Grained Top 100 Links Runned With All 3 Domains Fine Grained Top 500 Links With Lower PageRank . . . 51
3.7 All 3 Domains Fine Grained Top 100 Links Runned With Politics Fine Grained Top 500 Links 51
3.8 All 3 Domains Fine Grained Top 100 Links Runned With Politics Fine Grained Top 500 Links With Lower PageRank 51
3.9 All 3 Domains Fine Grained Top 100 Links Runned With Sport Fine Grained Top 500 Links 52
3.10 All 3 Domains Fine Grained Top 100 Links Runned With Sport Fine Grained Top 500 Links With Lower PageRank 52
3.11 All 3 Domains Fine Grained Top 100 Links Runned With Transportation Fine Grained Top 500 Links 52
3.12 All 3 Domains Fine Grained Top 100 Links Runned With Transportation Fine Grained Top 500 Links With Lower PageRank . . . 53

3.13 All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 500 Links 53
3.14 All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 500 Links With Lower PageRank . . . 54
3.15 All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 500 Links 54
3.16 All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 500 Links With Lower PageRank 54
3.17 All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 500 Links 55
3.18 All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 500 Links With Lower PageRank 55
3.19 All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 500 Links 55
3.20 All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 500 Links With Lower PageRank . . . 56
3.21 All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 10 Links 56
3.22 All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links 57
3.23 All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 10 Links 57
3.24 All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 100 Links 57
3.25 All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 10 Links 58
3.26 All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 100 Links 58
3.27 All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 10 Links 58
3.28 All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 100 Links 59
3.29 All 3 Domains Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 10 Links 59
3.30 All 3 Domains Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links 59
3.31 All 3 Domains Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links 60
3.32 All 3 Domains Fine Grained Top 500 Links Runned With Politics Fine Grained Top 10 Links 60
3.33 All 3 Domains Fine Grained Top 500 Links Runned With Politics Fine Grained Top 100 Links 60
3.34 All 3 Domains Fine Grained Top 500 Links Runned With Politics Fine Grained Top 300 Links 61

3.35 All 3 Domains Fine Grained Top 500 Links Runned With Sport Fine Grained Top 10 Links 61
3.36 All 3 Domains Fine Grained Top 500 Links Runned With Sport Fine Grained Top 100 Links 61
3.37 All 3 Domains Fine Grained Top 500 Links Runned With Sport Fine Grained Top 300 Links 62
3.38 All 3 Domains Fine Grained Top 500 Links Runned With Transportation Fine Grained Top 10 Links 62
3.39 All 3 Domains Fine Grained Top 500 Links Runned With Transportation Fine Grained Top 100 Links 62
3.40 All 3 Domains Fine Grained Top 500 Links Runned With Transportation Fine Grained Top 300 Links 63
3.41 Politics Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links 63
3.42 Politics Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links 64
3.43 Politics Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links 64
3.44 Sport Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links 65
3.45 Sport Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links 65
3.46 Sport Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links 66
3.47 Transportation Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links 66
3.48 Transportation Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links 67
3.49 Transportation Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links 67

List of Tables

2.1 Testing computer parameters 9

3.1 All 3 domain model with top 300 abstracts, tested with all 3 domain texts 16
3.2 All 3 domain model with top 300 abstracts, tested with Politics abstracts (test file) 16
3.3 All 3 domain model with top 300 abstracts, tested with Sport abstracts (test file) 17
3.4 All 3 domain model with top 300 abstracts, tested with Transportation abstracts (test file) 17
3.5 All 3 domain model with top 300 abstracts, tested with all 3 domain texts 18
3.6 All 3 domain model with top 300 abstracts, tested with Politics abstracts (test file) 18
3.7 All 3 domain model with top 300 abstracts, tested with Sport abstracts (test file) 19
3.8 All 3 domain model with top 300 abstracts, tested with Transportation abstracts (test file) 19
3.9 Politics domain model with top 300 abstracts, tested with Politics abstracts (test file) in coarse grained 19
3.10 Politics domain model with top 300 abstracts, tested with Politics abstracts (test file) in fine grained 20
3.11 Sport domain model with top 300 abstracts, tested with Sport abstracts (test file) in coarse grained 20
3.12 Sport domain model with top 300 abstracts, tested with Sport abstracts (test file) in fine grained 20
3.13 Transportation domain model with top 300 abstracts, tested with Transportation abstracts (test file) in fine grained 20
3.14 Transportation domain model with top 300 abstracts, tested with Transportation abstracts (test file) in fine grained 21

3.15 TABLE 21
3.16 TABLE 22
3.17 TABLE 22
3.18 TABLE 22
3.19 TABLE 23
3.20 TABLE 23
3.21 TABLE 23
3.22 TABLE 24
3.23 TABLE 24
3.24 TABLE 24
3.25 TABLE 25
3.26 TABLE 25
3.27 TABLE 25
3.28 TABLE 25
3.29 TABLE 26
3.30 TABLE 26
3.31 TABLE 26
3.32 TABLE 27
3.33 TABLE 27
3.34 TABLE 28
3.35 TABLE 28
3.36 TABLE 28
3.37 TABLE 29
3.38 TABLE 29
3.39 TABLE 29
3.40 TABLE 29
3.41 TABLE 30
3.42 TABLE 30
3.43 TABLE 30
3.44 TABLE 30
3.45 TABLE 30
3.46 TABLE 31
3.47 TABLE 31
3.48 TABLE 31
3.49 TABLE 31
3.50 TABLE 32
3.51 TABLE 32
3.52 TABLE 32
3.53 TABLE 32
3.54 TABLE 32
3.55 TABLE 33
3.56 TABLE 33
3.57 TABLE 33
3.58 TABLE 33

3.59 TABLE 33
3.60 TABLE 33
3.61 TABLE 34
3.62 TABLE 34
3.63 TABLE 34
3.64 TABLE 35
3.65 TABLE 35
3.66 TABLE 35
3.67 TABLE 35
3.68 TABLE 36
3.69 TABLE 36
3.70 TABLE 36
3.71 TABLE 36
3.72 TABLE 37
3.73 TABLE 37
3.74 TABLE 37
3.75 TABLE 38
3.76 TABLE 38
3.77 TABLE 39
3.78 TABLE 39
3.79 TABLE 39
3.80 TABLE 40
3.81 TABLE 40
3.82 TABLE 40
3.83 TABLE 40
3.84 TABLE 40
3.85 TABLE 41
3.86 TABLE 41
3.87 TABLE 41
3.88 TABLE 41
3.89 TABLE 42
3.90 TABLE 42
3.91 TABLE 43
3.92 TABLE 43
3.93 TABLE 43
3.94 TABLE 44
3.95 TABLE 44
3.96 TABLE 44
3.97 TABLE 44
3.98 TABLE 44
3.99 All 3 Domains Fine Grained Top 300 With All 3 Domains Fine Grained Top 500 Links And All 3 Domains Fine Grained Top 500 Links With Lower PageRank 45

3.100All 3 Domains Fine Grained Top 500 Links With All 3 Domains Fine Grained Top 500 Links And All 3 Domains Fine Grained Top 500 Links With Lower Page Rank 46
3.101All 3 Domains Fine Grained Top 500 Links With All 3 Domains Fine Grained Top 500 Links With Lower PageRank 47
3.102Transportation Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links And Transportation Fine Grained Top 300 Links 48

Introduction

Motivation

Most Named Entity Recognition NER applications are trained on a general texts and on a specific domain, the problem is that they are optimized for specific type of data i.e. specific domain. That means that those NER applications can give a nice results on texts or domains that is trained, but bad results for texts on a specific domain for which that NER is not trained.

Most of the NER applications are trained on small number of types. For example Stanford NER has a model that have maximum 7 types.

Main goal of this thesis is to research/explore possibilities of training NER models for a specific domain. To achieve this goal it is necessary to create datasets for a certain domains. This research is focused on 3 domains, "POLITICS", "SPORT" and "TRANSPORTATION". Every domain is created with a certain number on types from DBpedia Ontology, then for creating a datasets is used DBpedia NIF who gives and opportunity to approaches to information from Wikipedia abstracts, for example types that annotated words has in those abstracts.

Thesis research which is the quality of trained domains, the impact of the size of the data and the quality of the defined domains.

Goals of the thesis

The main goal of the thesis is to create a domain specific models and try to get closer, or event better F1-score to the human F1-score. Then creating a smaller and larger models and their impact of scoring based on main model. And finally creating a coarse grained and fine grained models and monitoring their F1-score.

Thesis outline

Chapter 1

Background and related work

1.1 Information extraction

Information extraction first appears in late 1970s within NLP field. Based on Wikipedia[1], Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most of the cases this activity concerns processing human language texts by means of natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as information extraction. [https://en.wikipedia.org/wiki/Information_extraction] [1]
Another view of that what Information extraction is that automatically building a relational database from information contained in unstructured text. Unlike linear-chain models, general CRFs can capture long distance dependencies between labels. [http://people.cs.umass.edu/ mccallum/papers/crf-tutorial.pdf]
To understand better what IE is we can use example from [https://ontotext.com/knowledgehub/fundan extraction/], extracting information for mail message and adding to your Calendar. Millions of people use this on their daily basis and they are not aware of that how that works and what technology is used for that.
Figure 1.1 gives us closer look of what Information extraction (IE) is, and how State-of-the-Art algorithms transforms unstructured text to structured sequences understandable for machines.

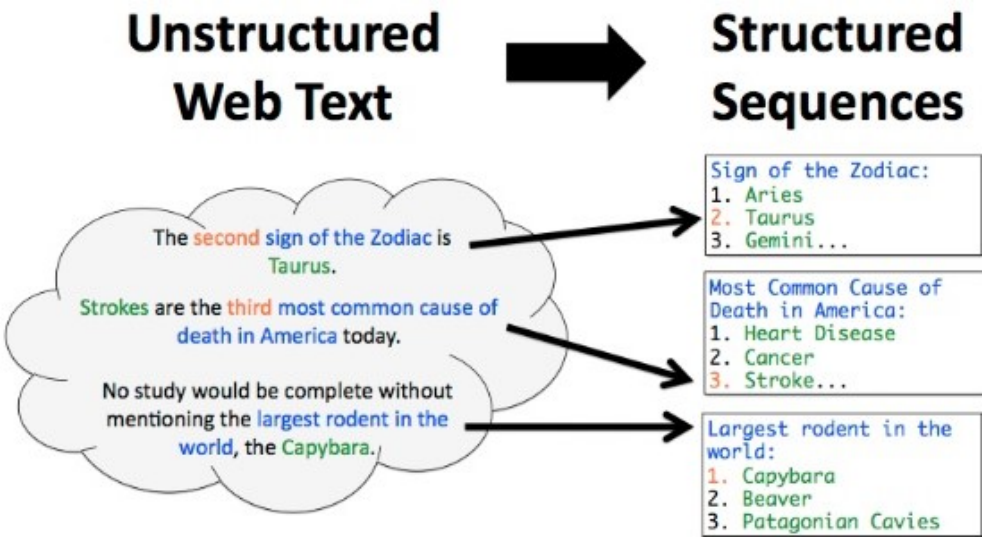


Figure 1.1: Information extraction, downloaded from https://www.slideshare.net/rubenizquierdobevia/information-extraction-45392844

1.2 Named Entity Recognition

Named Entity Recognition (NER) is the problem of identifying and classifying proper names in text, including locations, such as China; people, such as George Bush; and organizations, such as the United Nations. The named-entity recognition task is, given a sentence, first to segment which words are part of entities, and then to classify each entity by type (person, organization, location, and so on). The challenge of this problem is that many named entities are too rare to appear even in a large training set, and therefore the system must identify them based only on context. One approach to NER is to classify each word independently as one of either Person, Location, Organization, or Other (meaning not an entity). The problem with this approach is that it assumes that given the input, all of the named entity labels are independent. In fact, the named-entity labels of neighboring words are dependent; for example, while New York is a location, New York Times is an organization. [http://people.cs.umass.edu/ mccallum/papers/crf-tutorial.pdf section 1.2.2.2]

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction (IE) that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

Most research on NER systems has been structured as taking an unannotated block of text, such as this one:

Jim bought 300 shares of Acme Corp. in 2006.

And producing an annotated block of text that highlights the names of entities:

[Jim]Person bought 300 shares of [Acme Corp.]Organization in [2006]Time.

In this example, a person name consisting of one token, a two-token company name and a temporal expression have been detected and classified. [https://en.wikipedia.org/wiki/Named-entity_recognition]
Figure 1.2 shows how one NER application can look like. The text in example is predefined in Stanford NER application and loaded model (Classifier) is also trained by Stanford [https://nlp.stanford.edu/software/CRF-NER.html#Models].

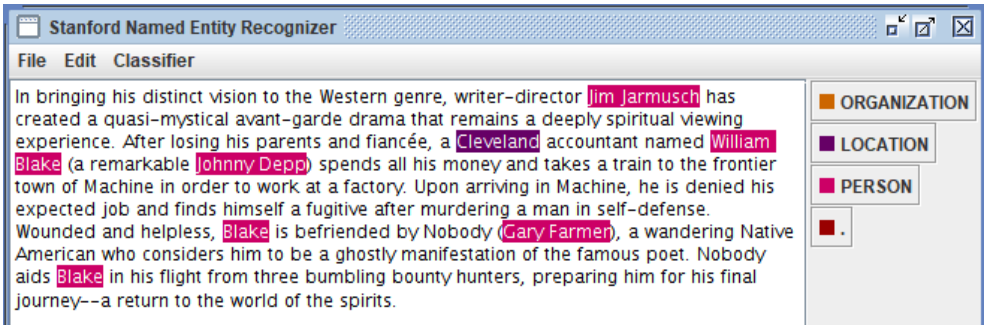


Figure 1.2: Stanford NER GUI with 3 classes model (Location, Person, Organization)

1.2.1 F_1 score

The success of NER systems is exposed to F_1 score (F-score or F-measure). F_1 score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. Written in formula, the $F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$ [https://en.wikipedia.org/wiki/F1_score]

1.2.2 Stanford NER

Stanford NER is a Java implementation of a Named Entity Recognizer. Named Entity Recognition (NER) labels sequences of words in a text which are the

names of things, such as person and company names, or gene and protein names. It comes with well-engineered feature extractors for Named Entity Recognition, and many options for defining feature extractors. Included with the download are good named entity recognizers for English, particularly for the 3 classes (PERSON, ORGANIZATION, LOCATION), and we also make available on this page various other models for different languages and circumstances, including models trained on just the CoNLL 2003 English training data.

Stanford NER is also known as CRFClassifier. The software provides a general implementation of (arbitrary order) linear chain Conditional Random Field (CRF) sequence models. That is, by training your own models on labeled data, you can actually use this code to build sequence models for NER or any other task.[https://nlp.stanford.edu/software/CRF-NER.html]

1.3 RDF/NIF

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications. [https://en.wikipedia.org/wiki/Resource_Description_Framework] More precisely RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. [https://www.w3.org/RDF/] Natural Language Processing Interchange Format (NIF) is an RDF-based format. The classes to represent linguistic data are defined in the NIF Core Ontology. All ontology classes are derived from the main class nif:String which respresents strings of Unicode characters.[https://www.w3.org/2015/09/bpmlod-reports/nif-based-nlp-webservices/#natural-language-%processing-interchange-format-nif] [http://aksw.org/Projects/NIF.html].

1.4 Domain specific Named Entity Recognition

Traditionally Named Entity Recognition (NER) systems have been built using available annotated datasets (like CoNLL, MUC) and demonstrate excellent performance. However, these models fail to generalize onto other domains like Sports and Finance where conventions and language use can differ significantly. Furthermore, several domains do not have large amounts of annotated labeled data for training robust Named Entity Recognition models. [https://arxiv.org/pdf/1612.00148.pdf] With specifying the domain we can

create a bigger model with more annotated words and reading the whole text will be same or even faster that reading text with a global domain.

1.5 DBpedia ontology

The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ontology currently covers 685 classes which form a subsumption hierarchy and are described by 2,795 different properties.

Since the DBpedia 3.7 release, the ontology is a directed-acyclic graph, not a tree. Classes may have multiple superclasses, which was important for the mappings to schema.org. A taxonomy can still be constructed by ignoring all superclasses except the one that is specified first in the list and is considered the most important. [http://wiki.dbpedia.org/services-resources/ontology]

Dbpedia ontology classes can be found here [http://mappings.dbpedia.org/server/ontology/classes/

The DBpedia Ontology currently contains about 4,233,000 instances. Figure 1.3 shows the number of instances for several classes within the ontology. [http://wiki.dbpedia.org/services-resources/ontology]

Class	Instances
Resource (overall)	4,233,000
Place	735,000
Person	1,450,000
Work	411,000
Species	251,000
Organisation	241,000

Figure 1.3: Dbpedia Ontology - Instances per class

1.6 Evaluation/training datasets

For the aim of our experiments we have trained 57 datasets. For training we've used Stanford NER application explained in subsection 1.2.2. We have two types of datasets, coarse-grained and fine-grained, also those types are divided in to specific domains and a global domain. To give an illustration, for

model with 100 retrieved abstract we will have 4 coarse-grained models (global domain and 3 specific domains), and similarly for a fine-grained models.

Wired links...

1.7 HDT

1.8 Apache Jena

Chapter 2

Domain specific named entity recognition

There are parameters of the computer used for tests shown in Table 2.1.

Table 2.1: Testing computer parameters

Part	Description
CPU	2.00 GHz Intel(R) Core(TM) i5-4310U
MEM	16 GB DDR3L
OS	x86_64 Windows 10 Pro
DISK	240GB SSD Kingston

2.1 Data pre-processing

First of all we downloaded 2 datasets from Dbpedia NIF Datasets [http://wiki.dbpedia.org/dbpedia-nif-dataset] for English language in .ttl format. Those datasets are:

- nif-context (or nif-abstract-context): the full text of a page as context (including begin and end index)
- nif-text-links: all in-text links to other DBpedia resources as well as external references

Another dataset that we needed was Dbpedia instance types dataset, downloaded from here [http://wiki.dbpedia.org/downloads-2016-04] also in .ttl format. This dataset contains all types from nif-text-links that occurrence at nif-abstract-context. So how all this dataset are connected between themselves? Let say that we have abstract for Alexander the Great. In nif-text-links file we have all words from abstract that has annotation, but we still don't know their type. So here comes instance types file where based on link from nif-text-link (eg.http://dbpedia.org/resource/Philip_II_of_Macedon) we can find

the type of annotated word (word Philip II has ontology type Monarch), but of coarse there can be a case that some words cannot be found on instance types file and automaticly have no type, or in our case ontology type O (O like Other). But know, let us explain deeply how we process and clean data from dataset. First we define small test dataset to check how fast we can process data. Running that dataset on downloaded files without any cleaning on data takes too long. So we said that converting the datasets from .ttl to .hdt (binary format) with rdf/hdt tool [http://www.rdfhdt.org/] will be faster. So we converted the datasets and reruned the algorithm again. There were some improvements, but not satisfying for our purposes. Our next solution was to clean datasets from unused data for our aims. The final result after cleaning the datasets was a smaller datasets, for instance nif-abstract-context file from 7.78GB now has 2.99GB, another big improvement was nif-text-links file who is reduced to 10.5GB from 44.6GB and at the end we also clean instance types file, but here we don't record any mayor memory improvements. Again we rerun the algorithm, of course there were improvements, but as well as previous the time that algorithm runs, was not acceptable for us. To give an illustration the time needed to find all types from one abstract in worst case, to read nif-text-links and instance type files until end was around 3.5 minutes. Therefore once more we converted our cleaned datasets from RDF format(.ttl) to binary format(.hdt). And how in previous running there were again improvements, but those improvements doesn't fulfill our expectations. The final think that we have to save us was creating a dataset tree. For nif-text-links file we created a tree where we have folders from "a-z", also special characters folders and other folder(this folder contains data that have a lower occurrence, let say & character or letters that are not part of English alphabet) and folders from "a-z" has subfolders also from a-z. To give a closer look how we create that tree, let say that we have an abstract for Volkswagen Golf MK3, so the link for that abstract would be http://dbpedia.org/resource/Volkswagen_Golf_Mk3 and this link will be stored to "v" folder and "o" subfolder. With this we have a smaller dataset where we can read whole one very fast.

For instance types file we modified the algorithm for creating a data tree. Here because of lower range data we have created only files from "a-z", of coarse special characters files and other file.

Finally we rerun the algorithm, and the time to process one abstract at worst case takes no more that 1 minute. Now we were ready to take next steps to retrieve types (subsection 2.3) and create domains (section 2.2) and prepare data for Stanford NER (section 2.4).

2.2 Domain specification

As we said earlier most of the NER application are trained on same domains, like ”PERSON”, ”ORGANIZATION” and ”LOCATION”. These 3 domains are widely spread all over the applications and perform nice results on text from this domains. So what we need is something that is not already trained or there is a small usage of that domain. After some research we find out that ”TRANSPORTATION” domain is not popular domain for NER applications, respectively in time of writing the thesis we don't find any usage of this domain. We see the possibility to create this specific domain. Types that we retrieve for this domain and specifying them are more deeply explained in Types retrieval section 2.3 in last paragraph. We have our first domain, but at least 2 more domains are needed to be able to make some experiments and conclusion.

Ideally will be those domains to have some connection between them and again to not be already wide spread. So we look at ontology types that are retrieved for ”TRANSPORTATION” domain, and there are types like Airport, Bridge, MetroStation and so on. This indicates to us that next domain can be ”POLITICS”. Why? Because some airports, bridges or metro stations bear names of Politicians. For instance airport in Prague, Czech Republic, is named by last president of Czechoslovakia, Vaclav Havel. Or another example is that some bridges in United States are named by famous politicians, like Presidents. Second domain is chosen, so we need at least one more domain to keep up with other NER applications. Again our reference was retrieved types for ”TRANSPORTATION” domain and.....

2.3 Types retrieval

After we solved the problem of that how effectively run the algorithm to find all types from the abstract, next issue was which types we want to be part of our domains and also which types we want to retrieve from Dbpedia. Worth mentioning that we will use the same ontology types for retrieving the abstracts links from Dbpedia and creating a domain models. For example the type ”Politician” will be used to retrieve links from Dbpedia that has that type, and also ”Politician” type will be use to annotated words, for instance Barack Obama will have type of ”Politician” (we will give more details on section 2.4).

In Dbpedia ontology classes page [http://mappings.dbpedia.org/server/ontology/classes/] we can see all types that Dbpedia has. Those ontology types are the same in instance types file also. Now we are facing with the fact that if we choose very small group of ontology types, at the experiment point we will have minor range of annotated words and experiments won't be relevant. On the other hand, if we go too deep to ontology types, we will have a lot of annotated

words, but training the model will take a lot of time and memory, and there is a possibility that we will reach memory exception, or because of big group of types training will never end.

After some testing with the number of retrieved types we finally found the best selection of types, in total we choose 275 ontology types for all domains.

Now let us explain more deeply every single domain and which types has that domain. We have 3 domains (see section 2.2 for that how we choose those domains) "POLITICS", "SPORT" and "TRANSPORTATION".

In "POLITICS" domain we retrieve in total 26 types (Parliament, Election, PoliticalParty, GeopoliticalOrganisation, Politician, Ambassador, Chancellor, Congressman, Deputy, Governor, Lieutenant, Mayor, MemberOfParliament, Minister, President, PrimeMinister, Senator, VicePresident, VicePrimeMinister, PoliticianSpouse, PersonFunction, PoliticalFunction, Profession, TopicalConcept and PoliticalConcept), which we sort in 11 more specific types like Ambassador, Chancellor, Congressman, Deputy, Governor, Lieutenant, Mayor, MemberOfParliament, Minister, President, PrimeMinister, Senator, VicePresident and VicePrimeMinister are joined together in one specific domain Politician, other types we leaved as it is, because if we group them the typeswon't give any sense. Although this domain has the lowest number of retrieved types, on creating a models this domain has the largest annotated words.

We do the same for "SPORT" domain where we retrieve in total 171 types (Sport, Athletics, currentWorldChampion, firstOlympicEventfootedness, TeamSport, SportsClub, HockeyClub, RugbyClub, SoccerClub, chairmanTitle, clubsRecordGoalscorer, fansgroup, firstGameground, largestWinmanagerTitle, worstDefeat, NationalSoccerClub, SportsLeague, AmericanFootballLeague, AustralianFootballLeague, AutoRacingLeague, BaseballLeague, BasketballLeague, BowlingLeague, BoxingLeague, CanadianFootballLeague, CricketLeague, CurlingLeague, CyclingLeague, FieldHockeyLeague, FormulaOneRacing, GolfLeague, HandballLeague, IceHockeyLeague, InlineHockeyLeague, LacrosseLeague, MixedMartialArtsLeague, MotorcycleRacingLeague, PaintballLeague, PoloLeague, RadioControlledRacingLeague, RugbyLeague, SoccerLeague, SoftballLeague, SpeedwayLeague, TennisLeague, VideogamesLeague, VolleyballLeague, SportsTeam, AmericanFootballTeam, AustralianFootballTeam, BaseballTeam, BasketballTeam, CanadianFootballTeam, CricketTeam, CyclingTeam, FormulaOneTeam, HandballTeam, HockeyTeam, SpeedwayTeam, Athlete, ArcherPlayer, AthleticsPlayer, AustralianRulesFootballPlayer, BadmintonPlayer, BaseballPlayer, BasketballPlayer, Bodybuilder, Boxer, AmateurBoxer, BullFighter, Canoeist, ChessPlayer, Cricketer, Cyclist, DartsPlayer, Fencer, GaelicGamesPlayer, GolfPlayer, GridironFootballPlayer, AmericanFootballPlayer, CanadianFootballPlayer, Gymnast, HandballPlayer, HighDiver, HorseRider, Jockey, LacrossePlayer, MartialArtist, MotorsportRacer, MotorcycleRider, MotocycleRacer, SpeedwayRider, RacingDriver, DTMRacer, FormulaOneRacer, NascarDriver, RallyDriver, NationalCollegiateAthleticAssociationAthlete, NetballPlayer, Pok-

erPlayer, Rower, RugbyPlayer, SnookerPlayer, SnookerChamp, SoccerPlayer, SquashPlayer, Surfer, Swimmer, TableTennisPlayer, TeamMember, TennisPlayer, VolleyballPlayer, BeachVolleyballPlayer, WaterPoloPlayer, WinterSportPlayer, Biathlete, BobsleighAthlete, CrossCountrySkier, Curler, FigureSkater, IceHockeyPlayer, NordicCombined, Skater, Ski_jumper, Skier, SpeedSkater, Wrestler, SumoWrestler, Coach, AmericanFootballCoach, CollegeCoach, VolleyballCoach, OrganisationMember, SportsTeamMember, SportsManager, SoccerManager, SportsEvent, CyclingCompetition, FootballMatch, GrandPrix, InternationalFootballLeagueEvent, MixedMartialArtsEvent, NationalFootballLeagueEvent, Olympics, OlympicEvent, Race, CyclingRace, HorseRace, MotorRace, Tournament, GolfTournament, SoccerTournament, TennisTournament, WomensTennisAssociationTournament, WrestlingEvent, SportCompetitionResult, OlympicResult, SnookerWorldRanking, SportsSeason, MotorsportSeason, SportsTeamSeason, BaseballSeason, FootballLeagueSeason, NationalFootballLeagueSeason, NCAATeamSeason, SoccerClubSeason, SoccerLeagueSeason, Referee, SportFacility, CricketGround, GolfCourse, RaceTrack and SkiArea), so those types, same as "POLITICS" domain, are specified in 8 types, like SportClub, SportsLeague, SportsTeam, Athlete, Coach, OrganizationMember, SportsManager and SportsEvent. This domain is a nice example of that even we retrieve quite a big number of types, we can reduce that number with more specific types which further don't lose the sense of type. For instance "David de Gea" has a type of SoccerPlayer, but after processing will have type of Athlete, which gives sense, because any type of sport player is an athlete.

At the end we repeat the process for "TRANSPORTATION" domain, where we retrieve in total 78 types (Aircraft, aircraftType, aircraftUser, ceiling, dischargeAverage, enginePower, engineType, gun, powerType, wingArea, wingspan, MilitaryAircraft, Automobile, automobilePlatform, bodyStyle, enginePower, engineType, powerType, transmission, Locomotive, boiler, boilerPressure, cylinderCount, enginePower, engineType, powerType, MilitaryVehicle, Motorcycle, On-SiteTransportation, ConveyorSystem, Escalator, MovingWalkway, Rocket, countryOrigin, finalFlight, lowerEarthOrbitPayload, maidenFlight, rocketFunction, rocketStages, Ship, captureDate, homeport, layingDown, maidenVoyage, numberOfPassengers, shipCrew, shipLaunch, SpaceShuttle, contractAward, Crews, firstFlight, lastFlight, missions, numberOfCrew, numberOfLaunchessatellitesDeployedSpaceStation, Spacecraft, cargoFuel, cargoGascargoWaterrocket, Train, locomotive, wagon, TrainCarriage, Tram, Engine, AutomobileEngine, RocketEngine, Company, PublicTransitSystem, Airline, BusCompany, Infrastructure, Airport, Port, RestArea, RouteOfTransportation, Bridge, RailwayLine, RailwayTunnel, Road, RoadJunction, RoadTunnel, WaterwayTunnel, Station, MetroStation, RailwayStation, RouteStop, TramStation) and minimized in 14 more specific types like Aircraft, Automobile, Locomotive, MilitaryVehicle, Motorcycle, On-SiteTransportation, Rocket, Ship, SpaceShuttle, SpaceStation, Spacecraft, Train, PublicTransit-

System and Infrastructure. The logic of that who we create more specific ontology types is same as in "POLITICS" or "SPORT" domain.

2.4 Data transformation/output for Stanford

We define domains as well their types that we will retrieve and process, now we should put everything together and prepare data for Stanford NER application[1.2.2]. In Data pre-processing section 2.1 we explain how we handled the data downloaded from web and we briefly touch how those data will be prepared for training in Stanford NER application.

We waned to get those links based in their PageRank[2]. In Java there is a framework called Apache Jena[3] that is used for working with RDF data and SPARQL queries. With a prepared and tested SPARQL queries on www.dbpedia.com/sparql we implemented retrieving link, on Java, on this endpoint. After retrieving those data, based on their order we search does retrieved link is contained on our abstract file. If link is found it's written to two files, one file is where are written all abstracts from every domain, and another file is file for that specific domain. Those files are creating in RDF format, with n-triples, that means that there is subject, in our case that is the link of abstract, then predicate who has isString annotation which tells that next triple contains the abstract text and finally object where abstract text is placed. Next think that we need to do is to find all annotated words from abstract and their types. The algorithm of finding types is explained in Section 2.1. What is not mention on that section is that after finding the types, the abstract is written to file, where on first position is word and on the second position is the type of that word, if there is any, if not the type is O. Final step is to prepare data to be able to train models in Stanford NER [1.2.2] with the types that we define in Section 2.3. Because files contains all types that were found on the abstracts we need to clean and group them, as well to create a coarse grained files. The algorithm is very simple, it reads the files who already has all types and if type is part of our retrieved types (Section 2.3) then either type is leaved as it is, or is grouped to more specific type, for instance if word has type Ambassador, then after filtering that word will have Politician type. The same is for coarse grained annotation, but here proper types after filtering has "POLITICS", "SPORT" or "TRANSPORTATION" type.

Chapter 3

Experiments

We have provide various types of experiments. In next sections we will discuss more about every provided experiment.

3.1 Goals of the experiments

We set a few goals of the experiments. First of all we waned to test does we will get better results if we run the model of all domains in coarse grained, against the model of all domains in fine grained. In this test we run the models also with all domains texts. Then we get those models and we run it with specific domain texts, in both fine and coarse grained. Also we make experiments with specific domain model runned with domain specific texts, for example, politics domain model in coarse grained is runned with politics domain text also annotated in coarse grained, politics domain model in fine grained is runned with politics domain text also annotated in fine grained, and the same for sport and transportation domains.

3.2 List of experiments

With our trained models we made a few experiments. First one is the model that has 300 abstract on every domain(900 abstract in total). This is our main model and other experiments that we will provide like models that has lower or higher number of abstracts or experiments where model has more abstracts that a test file or vice-verse, all those results will be compared with the results obtained from main experiment.

3.2.1 Main experiment

This is our main experiment where other experiments will be compared with this one. This model is trained with top 300 Wikipedia abstracts, for every domain, based on PageRank[2]. The text is annotated in coarse grained and

has around 2300 annotated words in total. First experiment that we do with this model is that we run it with the same text that model is created in coarse grained. Results are not bad at all, we are above 95% (see Table 3.2.1), which is great number for such middle weight model. But let see how model will behaves when we tested with abstracts for every domain.

Table 3.2.1 shows the output of model when is tested with abstracts from a "POLITICS" domain. As we said in Section 2.4 this type of abstract has the biggest word annotation. In this experiment, trained model annotated words with a "TRANSPORTATION" domain.

Table 3.2.1 gives us results from abstracts from "SPORT" domain. From the table we can see that our trained model annotated some words with a "POLITICS" or "TRANSPORTATION", even those that our test file contains only abstracts from "SPORT" domains and words has only "SPORT" type.

Table 3.2.1 provide outcome with testing with abstracts only from "TRANSPORTATION" domain. Also here the trained model annotated words with a "SPORT" type.

In conclusion with this kind of experiments we can say that it is not a good idea to train a model with all chosen domains and then use texts from every domain separately to perform NER.

Entity	**P**	**R**	**F1**
POLITICS	0,9872	0,9462	0,9662
SPORT	0,9846	0,9629	0,9736
TRANSPORTATION	0,9940	0,9823	0,9881
Totals	**0,9875**	**0,9625**	**0,9748**

Table 3.1: All 3 domain model with top 300 abstracts, tested with all 3 domain texts

Entity	**P**	**R**	**F1**
POLITICS	0,9839	0,4025	0,5713
TRANSPORTATION	0,0000	1,0000	0,0000
Totals	**0,9792**	**0,4025**	**0,5705**

Table 3.2: All 3 domain model with top 300 abstracts, tested with Politics abstracts (test file)

Entity	P	R	F1
POLITICS	0,0000	1,0000	0,0000
SPORT	0,9846	0,9628	0,9736
TRANSPORTATION	0,0000	1,0000	0,0000
Totals	**0,9819**	**0,9628**	**0,9722**

Table 3.3: All 3 domain model with top 300 abstracts, tested with Sport abstracts (test file)

Entity	P	R	F1
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	0,9940	0,9822	0,9880
Totals	**0,9861**	**0,9822**	**0,9841**

Table 3.4: All 3 domain model with top 300 abstracts, tested with Transportation abstracts (test file)

After we finish the experiments with model that is trained with all abstracts from every domain in coarse grained, we waned to see the impact of model that is trained with same abstracts, but now annotated in fine grained. Table 3.2.1 shows the results of provided experiment where we can see that we have a little bit more better results than experiment in Table 3.2.1.

Then we tested our model with abstracts from ”POLITICS” domain. How we can see from Table 3.2.1 there is some improvements on results, but no satisfying at all. Also table shows that some words are annotated with types from ”SPORT” and ”TRANSPORTATION” domain.

After that we rerun the experiment, but now with abstracts from ”SPORT” domain. In Table 3.2.1 we can see minor growth of the results unlike experiment in Table 3.2.1, but this improvements are so small that are almost unimportant. Also our model annotated some words with types from ”POLITICS” and ”TRANSPORTATION” domain which the test file don’t have those types at all.

Finally the last experiment with this model are the abstracts from ”TRANSPORTATION” domain. Table 3.2.1 shows the output of the provided experiment, where like in previous 2 experiments we can notice a very little improvements on results, from experiment in Table 3.2.1, who again can be unimportant.

Provided experiments with the model who is trained with all abstracts from every domain annotated in fine grained, overall provide a very little improvements on results on every experiment. With that observation trained model annotated in fine grained is better to use instead of the model that is annotated in coarse grained. Another benefit of this type of model is that we can see which types are annotated and their results.

Entity	P	R	F1
Aircraft	1,0000	1,0000	1,0000
Athlete	1,0000	0,9802	0,9900
Automobile	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,9820	0,9909
PoliticalParty	0,9860	0,9628	0,9743
Politician	1,0000	0,9353	0,9665
PublicTransitSystem	0,9919	0,9839	0,9879
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,9796	0,9683	0,9739
SportsEvent	1,0000	0,9242	0,9606
SportsLeague	0,9647	0,9805	0,9725
SportsManager	1,0000	0,9423	0,9703
SportsTeam	1,0000	0,9805	0,9902
Train	1,0000	1,0000	1,0000
Totals	**0,9880**	**0,9712**	**0,9795**

Table 3.5: All 3 domain model with top 300 abstracts, tested with all 3 domain texts

Entity	P	R	F1
Election	0,0000	0,0000	0,0000
PoliticalParty	0,9860	0,9628	0,9743
Politician	1,0000	0,1849	0,3120
PublicTransitSystem	0,0000	1,0000	0,0000
Ship	0,0000	1,0000	0,0000
SportsLeague	0,0000	1,0000	0,0000
Totals	**0,9825**	**0,4072**	**0,5758**

Table 3.6: All 3 domain model with top 300 abstracts, tested with Politics abstracts (test file)

Entity	**P**	**R**	**F1**
Athlete	1,0000	0,9802	0,9900
Coach	1,0000	1,0000	1,0000
Politician	0,0000	1,0000	0,0000
SportsClub	0,9794	0,9680	0,9737
SportsEvent	1,0000	0,9242	0,9606
SportsLeague	0,9678	0,9805	0,9741
SportsManager	1,0000	0,9423	0,9703
SportsTeam	1,0000	0,9804	0,9901
Train	0,0000	1,0000	0,0000
Totals	**0,9821**	**0,9716**	**0,9768**

Table 3.7: All 3 domain model with top 300 abstracts, tested with Sport abstracts (test file)

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Automobile	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,9820	0,9909
Politician	0,0000	1,0000	0,0000
PublicTransitSystem	0,9918	0,9837	0,9878
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,0000	1,0000	0,0000
SportsTeam	0,0000	1,0000	0,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9862**	**0,9881**	**0,9871**

Table 3.8: All 3 domain model with top 300 abstracts, tested with Transportation abstracts (test file)

Table 3.9: Politics domain model with top 300 abstracts, tested with Politics abstracts (test file) in coarse grained

Entity	**P**	**R**	**F1**
POLITICS	0,8039	0,6779	0,7355
Totals	**0,8039**	**0,6779**	**0,7355**

Table 3.10: Politics domain model with top 300 abstracts, tested with Politics abstracts (test file) in fine grained

Entity	**P**	**R**	**F1**
Election	0,8240	0,6398	0,7203
PoliticalParty	0,8100	0,7006	0,7513
Politician	0,8599	0,7234	0,7858
Totals	**0,8354**	**0,6980**	**0,7606**

Table 3.11: Sport domain model with top 300 abstracts, tested with Sport abstracts (test file) in coarse grained

Entity	**P**	**R**	**F1**
SPORT	0,9432	0,8839	0,9126
Totals	**0,9432**	**0,8839**	**0,9126**

Table 3.12: Sport domain model with top 300 abstracts, tested with Sport abstracts (test file) in fine grained

Entity	**P**	**R**	**F1**
Athlete	0,9713	0,8366	0,8989
Coach	1,0000	0,7500	0,8571
SportsClub	0,9453	0,9041	0,9242
SportsEvent	1,0000	0,7879	0,8814
SportsLeague	0,9418	0,8958	0,9182
SportsManager	1,0000	0,9615	0,9804
SportsTeam	0,9845	0,8301	0,9007
Totals	**0,9592**	**0,8750**	**0,9152**

Table 3.13: Transportation domain model with top 300 abstracts, tested with Transportation abstracts (test file) in fine grained

Entity	**P**	**R**	**F1**
TRANSPORTATION	0,9583	0,9109	0,9340
Totals	**0,9583**	**0,9109**	**0,9340**

Table 3.14: Transportation domain model with top 300 abstracts, tested with Transportation abstracts (test file) in fine grained

Entity	**P**	**R**	**F1**
Aircraft	0,9659	0,8333	0,8947
Automobile	1,0000	0,8000	0,8889
Infrastructure	0,9550	0,9550	0,9550
PublicTransitSystem	0,9662	0,9309	0,9482
Ship	1,0000	0,6429	0,7826
SpaceShuttle	1,0000	0,8333	0,9091
SpaceStation	0,0000	1,0000	0,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9660**	**0,9010**	**0,9324**

3.2.2 Experiments that have lower number of traning abstracts in model

In this experiment we get the model that is trained with all 3 domains together with coarse grained and we make an experiment with the text that is trained, that means text from all 3 domains annotated with coarse grained. Here we have for every domain 10 abstracts, so in total 30 annotated abstracts. This is our smallest model. How we can see from the picture in politics domain F1 value is lower than in sport and transportation. This results with lower F1 value in total, but still that is very satisfying value, because we have only 1 false positive and 2 false negative words.

Table 3.15: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9655	0,9333	0,9492
SPORT	1,0000	1,0000	1,0000
TRANSPORTATION	1,0000	1,0000	1,0000
Totals	**0,9811**	**0,9630**	**0,9720**

In this experiment we get the previous trained model and test with politics coarse grained text. Result is not satisfying, because we have a very low recall, so that results with bad F1 value and very big false negative words. We can clearly say that this is not the model that we want to use in annotating. With those results we can say that all 3 domains together with all text perform better results than all 3 domain model tested with specified domain.

Table 3.16: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9655	0,3636	0,5283
Totals	**0,9655**	**0,3636**	**0,5283**

Then we get the same model, and like previous, we tested with sport coarse grained text. So how we can see from the picture, we have perfect result, there is no false positive or false negative words, there are only true positive which results with F1 value of 1.000 (no lost words).

Table 3.17: TABLE

Entity	**P**	**R**	**F1**
SPORT	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Finally we tested this model with our last domain, transportation domain with coarse grained text. Because of very small annotated words we have again perfect results without any false positive or false negative words, same as previous test, there are only true positive matches, which is what we want.

Table 3.18: TABLE

Entity	**P**	**R**	**F1**
TRANSPORTATION	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

After we finish testing of all 3 domains trained with coarse grained texts, we move on with new type of experiment, model that contains all our domains and is trained with fine grained texts. We run the same experiment like in coarse grained, but now text is annotated with fine grained. So, in such small trained model is not surprising that we get the same results as in experiment with coarse grained annotated text.

Table 3.19: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
PoliticalParty	0,9600	0,9231	0,9412
Politician	1,0000	1,0000	1,0000
PublicTransitSystem	1,0000	1,0000	1,0000
Ship	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**0,9811**	**0,9630**	**0,9720**

We have repeated the same experiments from previous, but now with fine grained annotation. So we tested our model with politics domain annotated in fine grained and we get exactly the same results like in coarse grained. In this case there is no difference if we use coarse or fine grained trained model, because the results are same.

Table 3.20: TABLE

Entity	**P**	**R**	**F1**
Election	0,0000	0,0000	0,0000
PoliticalParty	0,9600	0,9231	0,9412
Politician	1,0000	1,0000	1,0000
Totals	**0,9655**	**0,3636**	**0,5283**

The sport domain text annotated in fine grained provide also same result as coarse grained annotated domain.

Table 3.21: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Same as sport domain, the transportation domain Figure **??** text annotated with fine grained, provide also same results.

Table 3.22: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
PublicTransitSystem	1,0000	1,0000	1,0000
Ship	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

After experimenting with models that are trained with all domains together, we move on with experiments where we create specific models, both coarse and fine grained. First we tested the model that is created with politics text annotated in coarse grained. How we can see from Figure **??** we have a better results than that experiment that we provide in all 3 domains. The F1 value is very big which is the results of low false positive and false negative annotated words.

Table 3.23: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9737	0,9610	0,9673
Totals	**0,9737**	**0,9610**	**0,9673**

We also provide the same experiment like previous, but now the model is annotated in fine grained. From Figure **??** is clearly that we have improve our model, here the F1 value is bigger that in coarse grained (see Figure **??**) and also false positive annotated words are decreased by 1. That means even we have a complete review of our entities and can see the precision of each entity, this domain, in total, is more precise that coarse grained model and all 3 domains in coarse and fine grained.

Table 3.24: TABLE

Entity	**P**	**R**	**F1**
Election	1,0000	0,9333	0,9655
PoliticalParty	0,9600	0,9231	0,9412
Politician	1,0000	1,0000	1,0000
Totals	**0,9867**	**0,9610**	**0,9737**

From here, with this small amount of data we can say that it's better to use a politics specific domain, in fine grained, than a global domain, because in global domain we have 28 words annotated as true positive, but in a politics specific coarse grained domain we have 74 words annotated as true positive, which is very big difference, if we want to be precise.

We repeated the experiment for sport specific domain, but in this case the results are exactly the same in coarse and fine grained (see Figure **??** and Figure **??** as well as in all 3 domains also in coarse and fine grained

Table 3.25: TABLE

Entity	**P**	**R**	**F1**
SPORT	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.26: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

This model in really small, only 15 annotated words, that's because we cannot make a conclusion which model is better to use. But if we speak about performance, of coarse the better solution is specified domain.

The transportation model, like a sport domain, is also a very small, only 9 annotated words. We provide the experiments for coarse grained (see Figure **??**) and we get same results as in all 3 domains (see Figure **??**). Same is happening for the fine grained domain (see Figure **??**). In such a small trained domain, we cannot clearly say which model is better to use.

Table 3.27: TABLE

Entity	**P**	**R**	**F1**
TRANSPORTATION	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.28: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
PublicTransitSystem	1,0000	1,0000	1,0000
Ship	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

In conclusion of this subsection, we can say that our strategy for creating a specific domain pays off, because in those domains we have a better results that in global domains. But let see in subsection **??** how domains will behave with a bigger data(text).

In subsection **??** we provide various experiments with different models. Now in this subsection we will repeat those experiments, but now we doubled the number of text to every domain, so we have 20 abstracts for every domain, in total 60 abstract. Those models, like previous, are trained in both coarse and fine grained. The first experiment in Figure **??** is now rerunned with a bigger model and text. From Figure **??** we can see that there are more annotated words and surprisingly better results than in previous domain. Then we tested the model with politics domain text in coarse grained annotation. Again the results are quite better than previous experiment (see Figure **??**), also now there is no false positive words, but this is still not satisfying. After that we tested this model with sport domain text in coarse grained annotation (see Figure **??**). And like in previous experiment (see Figure **??**) we have a perfect results, so we can say that this kind of model with very small domain annotated words is good to use. And finally we tested the model with transportation domain text in coarse grained annotation (see Figure **??**). Figure **??** show us that the model annotated some false positive word with SPORT, which even we have a perfect F1 value for transportation domain, in overall calculation, because of that word we have a lower F1 value than experiment in Figure **??**.

Table 3.29: TABLE

Entity	**P**	**R**	**F1**
POLITICS	1,0000	0,9615	0,9804
SPORT	1,0000	1,0000	1,0000
TRANSPORTATION	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,9780**	**0,9889**

Table 3.30: TABLE

Entity	**P**	**R**	**F1**
POLITICS	1,0000	0,3906	0,5618
Totals	**1,0000**	**0,3906**	**0,5618**

Table 3.31: TABLE

Entity	**P**	**R**	**F1**
SPORT	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.32: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	1,0000	1,0000	1,0000
Totals	**0,9231**	**1,0000**	**0,9600**

Similarly like experiment with a coarse grained model, we repeated the same experiments, but now, of coarse, with fine grained model. We test the trained model with text from all 3 domains together in fine grained annotation, and how we can see from Figure **??** the results are quite worst that in experiment with coarse grained (see Figure **??**), but better that the experiment where we have only 10 abstract (see Figure **??**). Furthermore we test the model with politics text, and how we can see from Figure **??** again results are not even close to previous experiment with all 3 domains together where we have better results like overall also in politics domain. Then we move on with a sport text (see Figure **??**) and again we have perfect results like in coarse grained (see Figure **??**) and the experiment with 10 abstract (see Figure **??**). And final experiment for this model is transportation text. Figure **??** shows us that results are worst that in coarse grained (see Figure **??**). Now we do not have any false positive word annotated with SPORT domain, but we have a lower number of annotated words which results with a lower F1 value in total.

Table 3.33: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,5000	0,6667
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,5000	0,6667
PoliticalParty	1,0000	0,9512	0,9750
Politician	1,0000	1,0000	1,0000
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,9560**	**0,9775**

Table 3.34: TABLE

Entity	**P**	**R**	**F1**
Election	0,0000	0,0000	0,0000
PoliticalParty	1,0000	0,9512	0,9750
Politician	1,0000	0,2000	0,3333
Totals	**1,0000**	**0,3906**	**0,5618**

Table 3.35: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.36: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,5000	0,6667
Infrastructure	1,0000	0,5000	0,6667
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsTeam	0,0000	1,0000	0,0000
Totals	**0,9091**	**0,8333**	**0,8696**

After we finish the experiments with models that are trained with texts from every domain, we do tests with models that are trained with texts from a particular domain (politics, sport and transportation). Figure **??** shows the output of politics coarse grain specified domain, where we can see that we have a way more better results than in experiment with a model trained with texts from all 3 domains (see Figure **??** and Figure **??**). The politics specific domain finds 125 true positive words unlike the experiments with all 3 domains that gives us only 50 true positive annotated words. However we repeated this experiment, but now with model trained in fine grained. Figure **??** shows the result of experiment, where we can see that there is no false positive annotated word like in previous experiment (see Figure **??**), which result with a better F1 value in total.

Table 3.37: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9921	0,9766	0,9843
Totals	**0,9655**	**0,3636**	**0,5283**

Table 3.38: TABLE

Entity	**P**	**R**	**F1**
Election	1,0000	0,9688	0,9841
PoliticalParty	1,0000	0,9512	0,9750
Politician	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,9766**	**0,9881**

Sport specific domain do not disappointed us. Figure **??** show the output of experiment in coarse grained and Figure **??** show the output of experiment in fine grained. Both experiment same as experiments with all 3 domains together (see Figure **??**, Figure **??**, Figure **??**, Figure **??**) give us perfect results. But if you want to be more precise, we recommend to use the specific model, because is fastest of reading words.

Table 3.39: TABLE

Entity	**P**	**R**	**F1**
SPORT	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.40: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

At the end we make experiments with a transportation specific domain. In Figure **??** we can see that our model gives us a worst results that experiment in all 3 domains. For instance experiment in Figure **??** gives us a perfect annotation with 1.0000 value at F1, also experiment in Figure **??** gives the same result like previous experiment, but here because of false positive annotated word in SPORT domain in total we have a lower value in F1. In overall experiments in all 3 domains gives a better result that the experiment in specific

domain. Of coarse we make an experiment with model that is trained with fine grained text. How Figure **??** shows that this model even gives worst result that the experiment with coarse grained model, and also worst results than the experiment in Figure **??**.

Table 3.41: TABLE

Entity	**P**	**R**	**F1**
TRANSPORTATION	1,0000	0,8333	0,9091
Totals	**1,0000**	**0,8333**	**0,9091**

Table 3.42: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,5000	0,6667
Infrastructure	1,0000	0,5000	0,6667
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,7500**	**0,8571**

Table 3.43: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9890	0,9375	0,9626
SPORT	1,0000	1,0000	1,0000
TRANSPORTATION	1,0000	0,9846	0,9922
Totals	**0,9960**	**0,9724**	**0,9841**

Table 3.44: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9890	0,3529	0,5202
Totals	**0,9890**	**0,3529**	**0,5202**

Table 3.45: TABLE

Entity	**P**	**R**	**F1**
SPORT	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.46: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	1,0000	0,9846	0,9922
Totals	**0,9697**	**0,9846**	**0,9771**

Table 3.47: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
Infrastructure	1,0000	1,0000	1,0000
PoliticalParty	0,9863	0,9730	0,9796
Politician	1,0000	0,8182	0,9000
PublicTransitSystem	1,0000	1,0000	1,0000
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**0,9960**	**0,9764**	**0,9861**

Table 3.48: TABLE

Entity	**P**	**R**	**F1**
Election	0,0000	0,0000	0,0000
PoliticalParty	0,9863	0,9730	0,9796
Politician	1,0000	0,1565	0,2707
Totals	**0,9890**	**0,3529**	**0,5202**

Table 3.49: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	1,0000	1,0000
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.50: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Infrastructure	1,0000	1,0000	1,0000
PublicTransitSystem	1,0000	1,0000	1,0000
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsTeam	0,0000	1,0000	0,0000
Totals	**0,9701**	**1,0000**	**0,9848**

Table 3.51: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9921	0,9804	0,9862
Totals	**0,9921**	**0,9804**	**0,9862**

Table 3.52: TABLE

Entity	**P**	**R**	**F1**
Election	1,0000	0,9848	0,9924
PoliticalParty	0,9863	0,9730	0,9796
Politician	1,0000	1,0000	1,0000
Totals	**0,9960**	**0,9882**	**0,9921**

Table 3.53: TABLE

Entity	**P**	**R**	**F1**
SPORT	1,0000	1,0000	1,0000
Totals	**1,0000**	**1,0000**	**1,0000**

Table 3.54: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	0,9474	0,9730
SportsEvent	1,0000	1,0000	1,0000
SportsLeague	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,9890**	**0,9945**

Table 3.55: TABLE

Entity	**P**	**R**	**F1**
TRANSPORTATION	1,0000	0,9846	0,9922
Totals	**1,0000**	**0,9846**	**0,9922**

Table 3.56: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Infrastructure	1,0000	1,0000	1,0000
PublicTransitSystem	1,0000	0,9630	0,9811
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsTeam	0,0000	1,0000	0,0000
Totals	**1,0000**	**0,9846**	**0,9922**

Table 3.57: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9920	0,9612	0,9764
SPORT	0,9963	0,9926	0,9944
TRANSPORTATION	1,0000	0,9735	0,9865
Totals	**0,9952**	**0,9766**	**0,9856**

Table 3.58: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9920	0,3615	0,5299
Totals	**0,9920**	**0,3615**	**0,5299**

Table 3.59: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,9962	0,9888	0,9925
Totals	**0,9962**	**0,9888**	**0,9925**

Table 3.60: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	1,0000	0,9735	0,9865
Totals	**0,9821**	**0,9735**	**0,9778**

Table 3.61: TABLE

Entity	P	R	F1
Aircraft	1,0000	0,6957	0,8205
Athlete	1,0000	0,4167	0,5882
Automobile	1,0000	1,0000	1,0000
Coach	1,0000	0,6667	0,8000
Infrastructure	1,0000	1,0000	1,0000
PoliticalParty	0,8774	0,6700	0,7598
Politician	1,0000	0,7455	0,8542
PublicTransitSystem	0,9744	0,7308	0,8352
Ship	1,0000	0,6000	0,7500
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,9512	0,9398	0,9455
SportsEvent	0,9737	0,8605	0,9136
SportsLeague	0,9500	0,8636	0,9048
SportsManager	1,0000	1,0000	1,0000
SportsTeam	1,0000	0,6364	0,7778
Train	1,0000	1,0000	1,0000
Totals	**0,9452**	**0,7535**	**0,8385**

Table 3.62: TABLE

Entity	P	R	F1
Election	0,0000	0,0000	0,0000
PoliticalParty	0,8774	0,6700	0,7598
Politician	1,0000	0,1285	0,2278
SportsEvent	0,0000	1,0000	0,0000
Totals	**0,8985**	**0,2580**	**0,4009**

Table 3.63: TABLE

Entity	P	R	F1
Athlete	1,0000	0,4167	0,5882
Coach	1,0000	0,6667	0,8000
SportsClub	0,9506	0,9390	0,9448
SportsEvent	1,0000	0,8605	0,9250
SportsLeague	0,9500	0,8636	0,9048
SportsManager	1,0000	1,0000	1,0000
SportsTeam	1,0000	0,6250	0,7692
Totals	**0,9683**	**0,7985**	**0,8753**

Table 3.64: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,6957	0,8205
Automobile	1,0000	1,0000	1,0000
Infrastructure	1,0000	1,0000	1,0000
PublicTransitSystem	0,9744	0,7308	0,8352
Ship	1,0000	0,6000	0,7500
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,0000	1,0000	0,0000
SportsTeam	0,0000	1,0000	0,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9677**	**0,7965**	**0,8738**

Table 3.65: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9956	0,9898	0,9927
Totals	**0,9956**	**0,9898**	**0,9927**

Table 3.66: TABLE

Entity	**P**	**R**	**F1**
Election	1,0000	0,9878	0,9939
PoliticalParty	0,9950	0,9852	0,9901
Politician	0,9937	0,9906	0,9922
Totals	**0,9956**	**0,9883**	**0,9920**

Table 3.67: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,9963	0,9963	0,9963
Totals	**0,9963**	**0,9963**	**0,9963**

Table 3.68: TABLE

Entity	P	R	F1
Athlete	1,0000	0,9722	0,9859
Coach	1,0000	1,0000	1,0000
SportsClub	1,0000	0,9878	0,9939
SportsEvent	1,0000	0,9767	0,9882
SportsLeague	1,0000	1,0000	1,0000
SportsManager	1,0000	1,0000	1,0000
SportsTeam	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,9888**	**0,9944**

Table 3.69: TABLE

Entity	P	R	F1
TRANSPORTATION	1,0000	0,9912	0,9956
Totals	**1,0000**	**0,9912**	**0,9956**

Table 3.70: TABLE

Entity	P	R	F1
Aircraft	1,0000	1,0000	1,0000
Automobile	1,0000	1,0000	1,0000
Infrastructure	1,0000	1,0000	1,0000
PublicTransitSystem	1,0000	0,9808	0,9903
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	0,6667	0,8000
SpaceStation	1,0000	1,0000	1,0000
Train	1,0000	1,0000	1,0000
Totals	**1,0000**	**0,9735**	**0,9865**

3.2.3 Experiments that have lower number of traning abstracts in model

Table 3.71: TABLE

Entity	P	R	F1
POLITICS	0,9804	0,9434	0,9615
SPORT	0,9832	0,9590	0,9709
TRANSPORTATION	0,9941	0,9754	0,9847
Totals	**0,9849**	**0,9584**	**0,9714**

Table 3.72: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9754	0,4082	0,5756
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	0,0000	1,0000	0,0000
Totals	**0,9531**	**0,4082**	**0,5716**

Table 3.73: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,0000	1,0000	0,0000
SPORT	0,9837	0,9588	0,9711
TRANSPORTATION	0,0000	1,0000	0,0000
Totals	**0,9805**	**0,9588**	**0,9695**

Table 3.74: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,0000	1,0000	0,0000
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	0,9939	0,9762	0,9806
Totals	**0,9861**	**0,9822**	**0,9841**

Table 3.75: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Athlete	1,0000	0,9899	0,9949
Automobile	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,9896	0,9948
PoliticalParty	0,9766	0,9486	0,9624
Politician	1,0000	0,9893	0,9946
PublicTransitSystem	0,9935	0,9776	0,9855
Ship	1,0000	0,9231	0,9600
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,9796	0,9658	0,9726
SportsEvent	1,0000	0,8636	0,9268
SportsLeague	0,9698	0,9835	0,9766
SportsManager	1,0000	0,9726	0,9861
SportsTeam	1,0000	0,9851	0,9925
Train	1,0000	1,0000	1,0000
Totals	**0,9870**	**0,9709**	**0,9789**

Table 3.76: TABLE

Entity	**P**	**R**	**F1**
Aircraft	0,0000	1,0000	0,0000
Election	0,0000	0,0000	0,0000
PoliticalParty	0,9766	0,9484	0,9623
Politician	1,0000	0,2092	0,3460
PublicTransitSystem	0,0000	1,0000	0,0000
Ship	0,0000	1,0000	0,0000
SportsClub	0,0000	1,0000	0,0000
SportsLeague	0,0000	1,0000	0,0000
Totals	**0,9619**	**0,4151**	**0,5799**

Table 3.77: TABLE

Entity	**P**	**R**	**F1**
Aircraft	0,0000	1,0000	0,0000
Athlete	1,0000	0,9899	0,9949
Coach	1,0000	1,0000	1,0000
PoliticalParty	0,0000	1,0000	0,0000
Politician	0,0000	1,0000	0,0000
SportsClub	0,9794	0,9654	0,9724
SportsEvent	1,0000	0,8636	0,9268
SportsLeague	0,9696	0,9834	0,9765
SportsManager	1,0000	0,9726	0,9861
SportsTeam	1,0000	0,9850	0,9924
Train	0,0000	1,0000	0,0000
Totals	**0,9821**	**0,9721**	**0,9770**

Table 3.78: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	1,0000	1,0000
Automobile	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,9896	0,9948
PoliticalParty	0,0000	1,0000	0,0000
Politician	0,0000	1,0000	0,0000
PublicTransitSystem	0,9934	0,9773	0,9853
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,0000	1,0000	0,0000
SportsTeam	0,0000	1,0000	0,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9866**	**0,9866**	**0,9866**

Table 3.79: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9866	0,9479	0,9669
Totals	**0,9866**	**0,9479**	**0,9669**

Table 3.80: TABLE

Entity	**P**	**R**	**F1**
Election	0,9975	0,9590	0,9779
PoliticalParty	0,9767	0,9530	0,9647
Politician	0,9977	0,9920	0,9948
Totals	**0,9906**	**0,9717**	**0,9810**

Table 3.81: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,9858	0,9676	0,9766
Totals	**0,9858**	**0,8676**	**0,9766**

Table 3.82: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	0,9731	0,9863
Coach	1,0000	1,0000	1,0000
SportsClub	0,9815	0,9715	0,9765
SportsEvent	1,0000	0,9091	0,9524
SportsLeague	0,9718	0,9787	0,9752
SportsManager	1,0000	0,9726	0,9850
SportsTeam	1,0000	0,9900	0,9796
Totals	**0,9865**	**0,9727**	**0,9796**

Table 3.83: TABLE

Entity	**P**	**R**	**F1**
TRANSPORTATION	0,9954	0,9747	0,9850
Totals	**0,9954**	**0,9747**	**0,9850**

Table 3.84: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,9835	0,9917
Automobile	1,0000	0,9583	0,9787
Infrastructure	1,0000	0,9948	0,9974
PublicTransitSystem	0,9934	0,9773	0,9853
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	0,6667	0,8000
SpaceStation	1,0000	1,0000	1,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9970**	**0,9807**	**0,9888**

Table 3.85: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9788	0,9444	0,9613
SPORT	0,9850	0,9596	0,9721
TRANSPORTATION	0,9962	0,9750	0,9855
Totals	**0,9857**	**0,9587**	**0,9720**

Table 3.86: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9734	0,4095	0,5765
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	0,0000	1,0000	0,0000
Totals	**0,9549**	**0,4095**	**0,5732**

Table 3.87: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,0000	1,0000	0,0000
SPORT	0,9849	0,9594	0,9720
TRANSPORTATION	0,0000	1,0000	0,0000
Totals	**0,9788**	**0,9594**	**0,9690**

Table 3.88: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,0000	1,0000	0,0000
SPORT	0,0000	1,0000	0,0000
TRANSPORTATION	0,9961	0,9769	0,9864
Totals	**0,9870**	**0,9769**	**0,9819**

Table 3.89: TABLE

Entity	P	R	F1
Aircraft	1,0000	0,9929	0,9964
Athlete	1,0000	0,9896	0,9948
Automobile	1,0000	1,0000	1,0000
Coach	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,9783	0,9890
PoliticalParty	0,9775	0,9403	0,9585
Politician	1,0000	0,9874	0,9937
PublicTransitSystem	0,9944	0,9807	0,9875
Ship	1,0000	0,9259	0,9615
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,9756	0,9553	0,9654
SportsEvent	1,0000	0,8796	0,9360
SportsLeague	0,9700	0,9810	0,9755
SportsManager	1,0000	0,9780	0,9889
SportsTeam	1,0000	0,9831	0,9915
Train	1,0000	1,0000	1,0000
Totals	**0,9866**	**0,9669**	**0,9766**

Table 3.90: TABLE

Entity	P	R	F1
Aircraft	0,0000	1,0000	0,0000
Election	0,0000	0,0000	0,0000
PoliticalParty	0,9774	0,9400	0,9583
Politician	1,0000	0,2171	0,3567
PublicTransitSystem	0,0000	1,0000	0,0000
Ship	0,0000	1,0000	0,0000
SportsClub	0,0000	1,0000	0,0000
SportsLeague	0,0000	1,0000	0,0000
Totals	**0,9631**	**0,4138**	**0,5789**

Table 3.91: TABLE

Entity	**P**	**R**	**F1**
Aircraft	0,0000	1,0000	0,0000
Athlete	1,0000	0,9896	0,9948
Coach	1,0000	1,0000	1,0000
PoliticalParty	0,0000	1,0000	0,0000
Politician	0,0000	1,0000	0,0000
SportsClub	0,9753	0,9549	0,9650
SportsEvent	1,0000	0,8796	0,9360
SportsLeague	0,9717	0,9810	0,9763
SportsManager	1,0000	0,9780	0,9889
SportsTeam	1,0000	0,9831	0,9915
Train	0,0000	1,0000	0,0000
Totals	**0,9785**	**0,9690**	**0,9737**

Table 3.92: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,9927	0,9963
Automobile	1,0000	1,0000	1,0000
Infrastructure	1,0000	0,9783	0,9890
PoliticalParty	0,0000	1,0000	0,0000
Politician	0,0000	1,0000	0,0000
PublicTransitSystem	0,9943	0,9804	0,9873
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	1,0000	1,0000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,0000	1,0000	0,0000
SportsTeam	0,0000	1,0000	0,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9884**	**0,9833**	**0,9858**

Table 3.93: TABLE

Entity	**P**	**R**	**F1**
POLITICS	0,9808	0,9450	0,9626
Totals	**0,9808**	**0,9450**	**0,9626**

Table 3.94: TABLE

Entity	**P**	**R**	**F1**
Election	0,9915	0,9393	0,9647
PoliticalParty	0,9777	0,9502	0,9637
Politician	0,9962	0,9877	0,9919
Totals	**0,9890**	**0,9648**	**0,9768**

Table 3.95: TABLE

Entity	**P**	**R**	**F1**
SPORT	0,9856	0,9706	0,9780
Totals	**0,9856**	**0,9706**	**0,9780**

Table 3.96: TABLE

Entity	**P**	**R**	**F1**
Athlete	1,0000	0,9791	0,9894
Coach	1,0000	1,0000	1,0000
SportsClub	0,9771	0,9630	0,9700
SportsEvent	1,0000	0,9074	0,9515
SportsLeague	0,9755	0,9848	0,9801
SportsManager	1,0000	0,9780	0,9889
SportsTeam	1,0000	0,9915	0,9957
Totals	**0,9861**	**0,9731**	**0,9796**

Table 3.97: TABLE

Entity	**P**	**R**	**F1**
TRANSPORTATION	0,9974	0,9756	0,9864
Totals	**0,9974**	**0,9756**	**0,9864**

Table 3.98: TABLE

Entity	**P**	**R**	**F1**
Aircraft	1,0000	0,9781	0,9889
Automobile	1,0000	0,8800	0,9362
Infrastructure	1,0000	0,9870	0,9934
PublicTransitSystem	0,9915	0,9804	0,9860
Ship	1,0000	1,0000	1,0000
SpaceShuttle	1,0000	0,6667	0,8000
SpaceStation	1,0000	1,0000	1,0000
Train	1,0000	1,0000	1,0000
Totals	**0,9961**	**0,9769**	**0,9864**

3.2.4 MIXED

Entity	P	R	F1
Aircraft	0,9242	0,5755	0,7093
Athlete	0,8182	0,3778	0,5169
Automobile	0,9565	0,3607	0,5238
Coach	1,0000	0,2000	0,3333
Infrastructure	1,0000	0,9896	0,9948
Locomotive	0,0000	0,0000	0,0000
Motorcycle	0,0000	0,0000	0,0000
OrganisationMember	0,0000	0,0000	0,0000
PoliticalParty	0,7656	0,5819	0,6613
Politician	0,8925	0,4099	0,5618
PublicTransitSystem	0,8291	0,6178	0,7080
Ship	0,9375	0,3409	0,5000
SpaceShuttle	1,0000	0,3750	0,5455
SpaceStation	1,0000	0,3333	0,5000
SportsClub	0,8009	0,4276	0,5575
SportsEvent	0,9559	0,3171	0,4762
SportsLeague	0,8071	0,5912	0,6824
SportsManager	0,9643	0,2903	0,4463
SportsTeam	0,8856	0,5838	0,7037
Train	1,0000	0,5455	0,7059
Totals	**0,8206**	**0,4837**	**0,6087**

Table 3.99: All 3 Domains Fine Grained Top 300 With All 3 Domains Fine Grained Top 500 Links And All 3 Domains Fine Grained Top 500 Links With Lower PageRank

Entity	P	R	F1
Aircraft	0,9735	0,6934	0,8099
Athlete	0,9101	0,6222	0,7391
Automobile	1,0000	0,4098	0,5814
Coach	1,0000	0,3000	0,4615
Infrastructure	0,8885	0,5218	0,6575
Locomotive	0,0000	0,0000	0,0000
Motorcycle	0,0000	0,0000	0,0000
OrganisationMember	0,0000	0,0000	0,0000
PoliticalParty	0,8393	0,7403	0,7876
Politician	0,9271	0,6593	0,7706
PublicTransitSystem	0,9027	0,7389	0,8126
Rocket	0,0000	0,0000	0,0000
Ship	0,9615	0,5682	0,7143
SpaceShuttle	1,0000	0,4375	0,6087
SpaceStation	1,0000	0,6667	0,8000
SportsClub	0,8722	0,6071	0,7159
SportsEvent	0,9000	0,4829	0,6286
SportsLeague	0,8622	0,7357	0,7939
SportsManager	0,9787	0,4946	0,6571
SportsTeam	0,9276	0,7514	0,8302
Train	1,0000	0,5455	0,7059
Totals	**0,8844**	**0,6592**	**0,7553**

Table 3.100: All 3 Domains Fine Grained Top 500 Links With All 3 Domains Fine Grained Top 500 Links And All 3 Domains Fine Grained Top 500 Links With Lower Page Rank

Entity	P	R	F1
Aircraft	0,6667	0,1111	0,1905
Athlete	0,4675	0,1268	0,1994
Automobile	0,0000	0,0000	0,0000
Coach	0,0000	0,0000	0,0000
Infrastructure	0,5352	0,1387	0,2203
Locomotive	0,0000	0,0000	0,0000
Motorcycle	0,0000	0,0000	0,0000
OrganisationMember	0,0000	0,0000	0,0000
PoliticalParty	0,5462	0,4097	0,4682
Politician	0,5962	0,1867	0,2844
PublicTransitSystem	0,6943	0,4098	0,5154
Rocket	0,0000	0,0000	0,0000
Ship	0,0000	0,0000	0,0000
SpaceShuttle	0,0000	0,0000	0,0000
SpaceStation	0,0000	0,0000	0,0000
SportsClub	0,6370	0,2675	0,3768
SportsEvent	0,2667	0,0412	0,0714
SportsLeague	0,6395	0,4125	0,5015
SportsManager	0,6000	0,0316	0,0600
SportsTeam	0,6316	0,2975	0,4045
Train	0,0000	0,0000	0,0000
Totals	**0,5983**	**0,2670**	**0,3692**

Table 3.101: All 3 Domains Fine Grained Top 500 Links With All 3 Domains Fine Grained Top 500 Links With Lower PageRank

Entity	P	R	F1
Aircraft	0,9950	0,9706	0,9826
Athlete	0,0000	0,0000	0,0000
Automobile	1,0000	0,8500	0,9189
Coach	0,0000	0,0000	0,0000
Infrastructure	1,0000	0,9820	0,9909
PoliticalParty	0,0000	0,0000	0,0000
Politician	0,0000	0,0000	0,0000
PublicTransitSystem	0,9835	0,9676	0,9755
Ship	1,0000	0,9655	0,9825
SpaceShuttle	1,0000	0,6667	0,8000
SpaceStation	1,0000	1,0000	1,0000
SportsClub	0,0000	0,0000	0,0000
SportsEvent	0,0000	0,0000	0,0000
SportsLeague	0,0000	0,0000	0,0000
SportsManager	0,0000	0,0000	0,0000
SportsTeam	0,0000	0,0000	0,0000
Train	1,0000	0,9091	0,9524
Totals	**0,9909**	**0,3491**	**0,5163**

Table 3.102: Transportation Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links And Transportation Fine Grained Top 300 Links

3.2.5 Experiments with lower training abstracts on model, but higher abstracts on test file

```
CRFClassifier tagged 97089 words in 1 documents at 2905,12 words per second.
        Entity P       R       F1      TP      FP      FN
      Aircraft 1,0000  0,0435  0,0833  1       0       22
       Athlete 1,0000  0,0278  0,0541  1       0       35
    Automobile 0,0000  0,0000  0,0000  0       0       3
         Coach 1,0000  0,3333  0,5000  1       0       2
Infrastructure 0,0000  0,0000  0,0000  0       0       22
 PoliticalParty 0,6458  0,3054  0,4147  62      34      141
     Politician 1,0000  0,0727  0,1356  4       0       51
PublicTransitSystem     0,8750  0,1346  0,2333  7       1       45
          Ship 1,0000  0,2000  0,3333  1       0       4
   SpaceShuttle 0,0000  0,0000  0,0000  0       0       6
   SpaceStation 0,0000  0,0000  0,0000  0       0       1
     SportsClub 0,9000  0,1084  0,1935  9       1       74
    SportsEvent 1,0000  0,0233  0,0455  1       0       42
   SportsLeague 0,6667  0,2121  0,3218  14      7       52
  SportsManager 0,0000  0,0000  0,0000  0       0       6
     SportsTeam 1,0000  0,0303  0,0588  1       0       32
          Train 0,0000  0,0000  0,0000  0       0       1
         Totals 0,7034  0,1591  0,2595  102     43      539

C:\Dev\stanford-ner-2017-06-09>java -cp stanford-ner.jar edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier modelsWit
hOneAnnotation\Top10Links\ner-All3DomainsTop10LinksFineGrained.ser.gz -testFile domainsWithOneAnnotation\Top100Links\All
3DomainsTop100LinksTextWithSameProcessedLinksOnAbstractFineGrainedSpecified.tsv
```

Figure 3.1: All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 100 Links

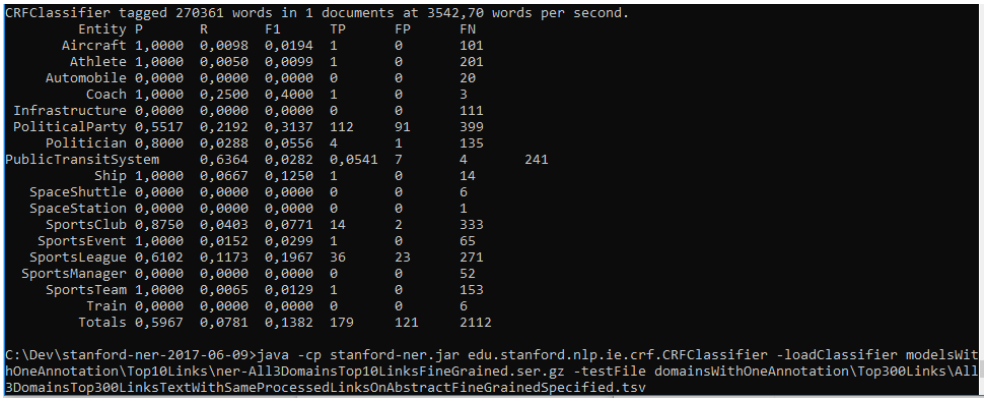


Figure 3.2: All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 300 Links

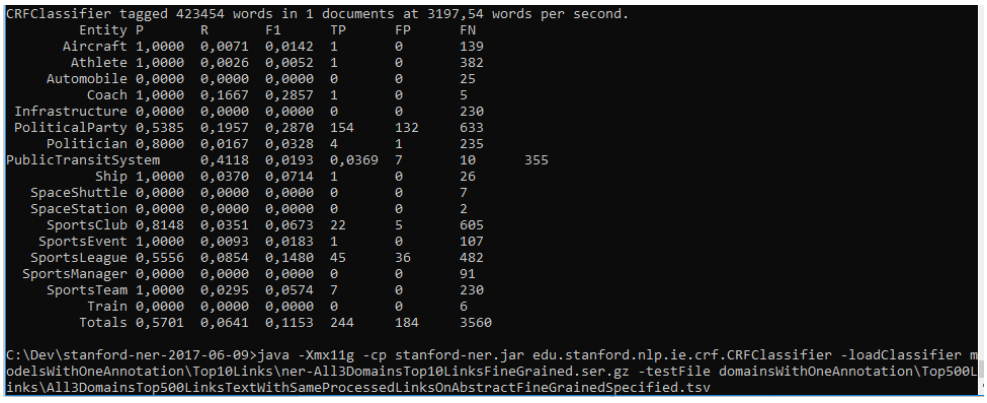


Figure 3.3: All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 500 Links

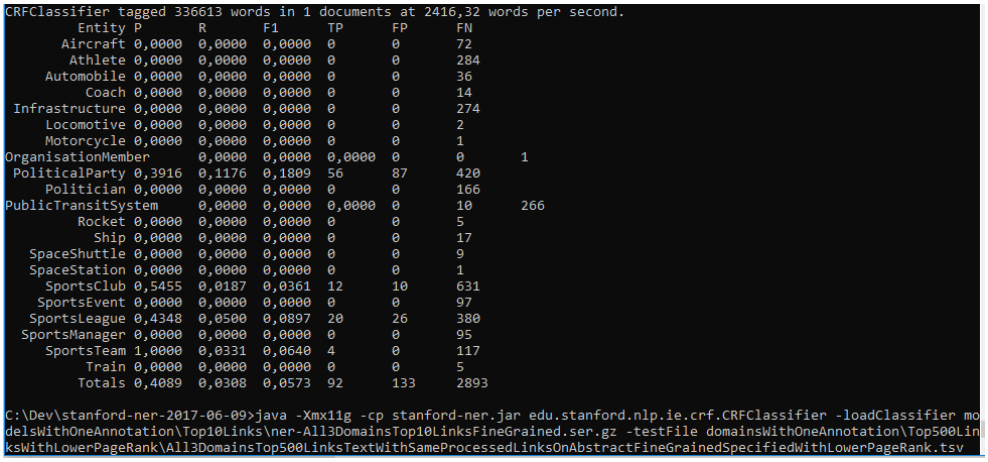


Figure 3.4: All 3 Domains Fine Grained Top 10 Links Runned With All 3 Domains Fine Grained Top 500 Links With Lower PageRank

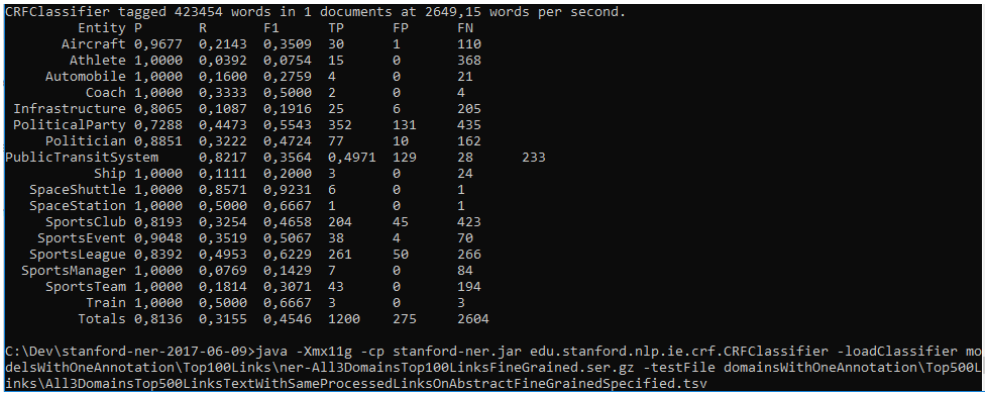


Figure 3.5: All 3 Domains Fine Grained Top 100 Links Runned With All 3 Domains Fine Grained Top 500 Links

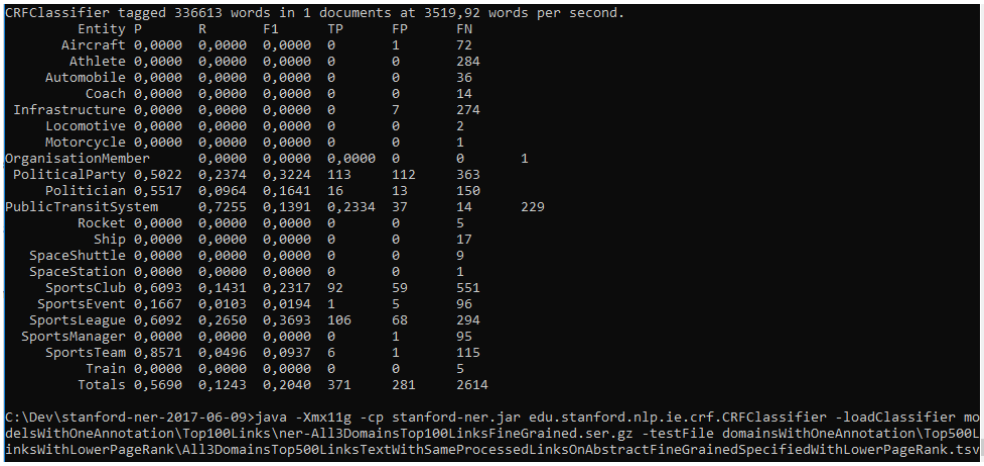


Figure 3.6: All 3 Domains Fine Grained Top 100 Links Runned With All 3 Domains Fine Grained Top 500 Links With Lower PageRank

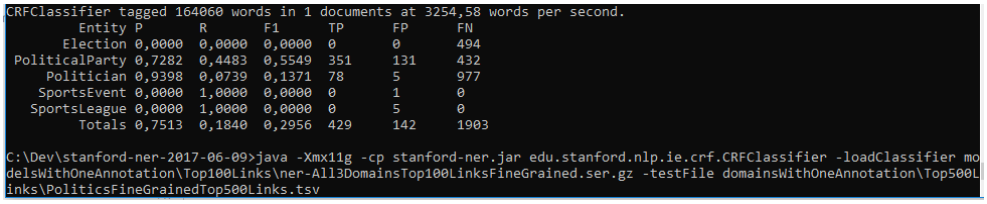


Figure 3.7: All 3 Domains Fine Grained Top 100 Links Runned With Politics Fine Grained Top 500 Links

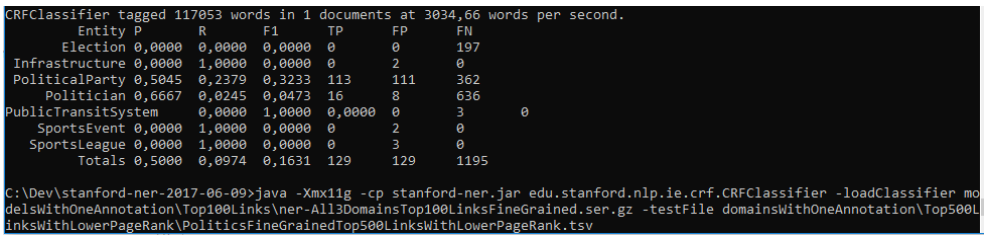


Figure 3.8: All 3 Domains Fine Grained Top 100 Links Runned With Politics Fine Grained Top 500 Links With Lower PageRank

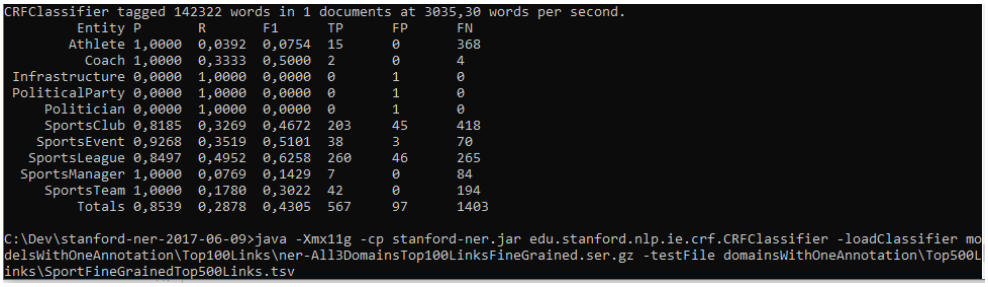


Figure 3.9: All 3 Domains Fine Grained Top 100 Links Runned With Sport Fine Grained Top 500 Links

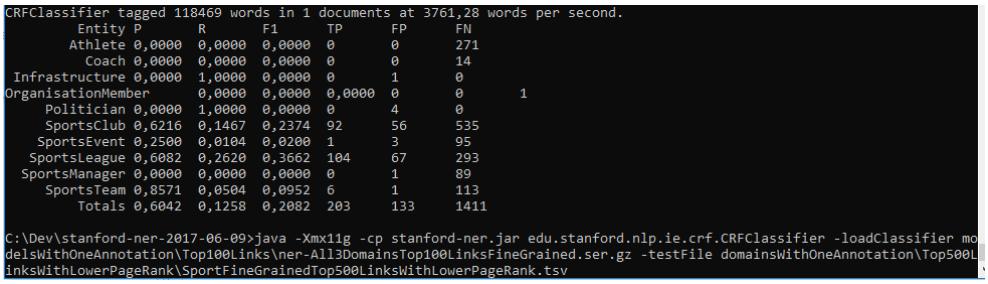


Figure 3.10: All 3 Domains Fine Grained Top 100 Links Runned With Sport Fine Grained Top 500 Links With Lower PageRank

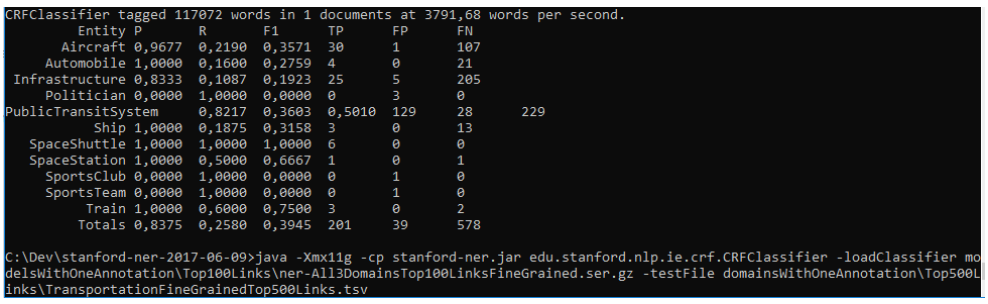
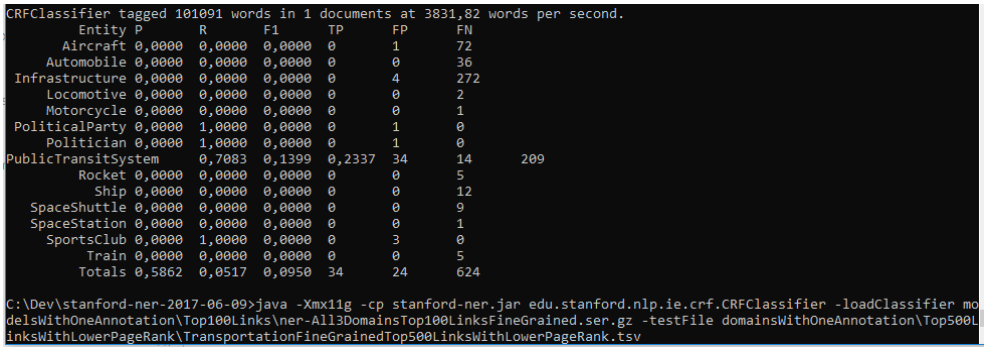


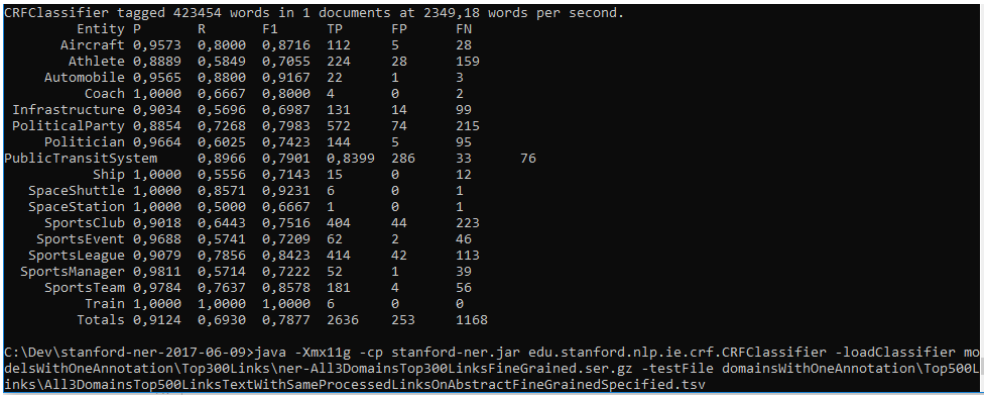
Figure 3.11: All 3 Domains Fine Grained Top 100 Links Runned With Transportation Fine Grained Top 500 Links



```
CRFClassifier tagged 101091 words in 1 documents at 3831,82 words per second.
         Entity P       R       F1      TP      FP      FN
       Aircraft 0,0000  0,0000  0,0000  0       1       72
     Automobile 0,0000  0,0000  0,0000  0       0       36
 Infrastructure 0,0000  0,0000  0,0000  0       4       272
     Locomotive 0,0000  0,0000  0,0000  0       0       2
     Motorcycle 0,0000  0,0000  0,0000  0       0       1
 PoliticalParty 0,0000  1,0000  0,0000  0       1       0
     Politician 0,0000  1,0000  0,0000  0       1       0
PublicTransitSystem     0,7083  0,1399  0,2337  34      14      209
         Rocket 0,0000  0,0000  0,0000  0       0       5
           Ship 0,0000  0,0000  0,0000  0       0       12
   SpaceShuttle 0,0000  0,0000  0,0000  0       0       9
   SpaceStation 0,0000  0,0000  0,0000  0       0       1
     SportsClub 0,0000  1,0000  0,0000  0       3       0
          Train 0,0000  0,0000  0,0000  0       0       5
         Totals 0,5862  0,0517  0,0950  34      24      624

C:\Dev\stanford-ner-2017-06-09>java -Xmx11g -cp stanford-ner.jar edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier mo
delsWithOneAnnotation\Top100Links\ner-All3DomainsTop100LinksFineGrained.ser.gz -testFile domainsWithOneAnnotation\Top500L
inksWithLowerPageRank\TransportationFineGrainedTop500LinksWithLowerPageRank.tsv
```

Figure 3.12: All 3 Domains Fine Grained Top 100 Links Runned With Transportation Fine Grained Top 500 Links With Lower PageRank



```
CRFClassifier tagged 423454 words in 1 documents at 2349,18 words per second.
         Entity P       R       F1      TP      FP      FN
       Aircraft 0,9573  0,8000  0,8716  112     5       28
        Athlete 0,8889  0,5849  0,7055  224     28      159
     Automobile 0,9565  0,8800  0,9167  22      1       3
          Coach 1,0000  0,6667  0,8000  4       0       2
 Infrastructure 0,9034  0,5696  0,6987  131     14      99
 PoliticalParty 0,8854  0,7268  0,7983  572     74      215
     Politician 0,9664  0,6025  0,7423  144     5       95
PublicTransitSystem     0,8966  0,7901  0,8399  286     33      76
           Ship 1,0000  0,5556  0,7143  15      0       12
   SpaceShuttle 1,0000  0,8571  0,9231  6       0       1
   SpaceStation 1,0000  0,5000  0,6667  1       0       1
     SportsClub 0,9018  0,6443  0,7516  404     44      223
    SportsEvent 0,9688  0,5741  0,7209  62      2       46
   SportsLeague 0,9079  0,7856  0,8423  414     42      113
  SportsManager 0,9811  0,5714  0,7222  52      1       39
     SportsTeam 0,9784  0,7637  0,8578  181     4       56
          Train 1,0000  1,0000  1,0000  6       0       0
         Totals 0,9124  0,6930  0,7877  2636    253     1168

C:\Dev\stanford-ner-2017-06-09>java -Xmx11g -cp stanford-ner.jar edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier mo
delsWithOneAnnotation\Top300Links\ner-All3DomainsTop300LinksFineGrained.ser.gz -testFile domainsWithOneAnnotation\Top500L
inks\All3DomainsTop500LinksTextWithSameProcessedLinksOnAbstractFineGrainedSpecified.tsv
```

Figure 3.13: All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 500 Links

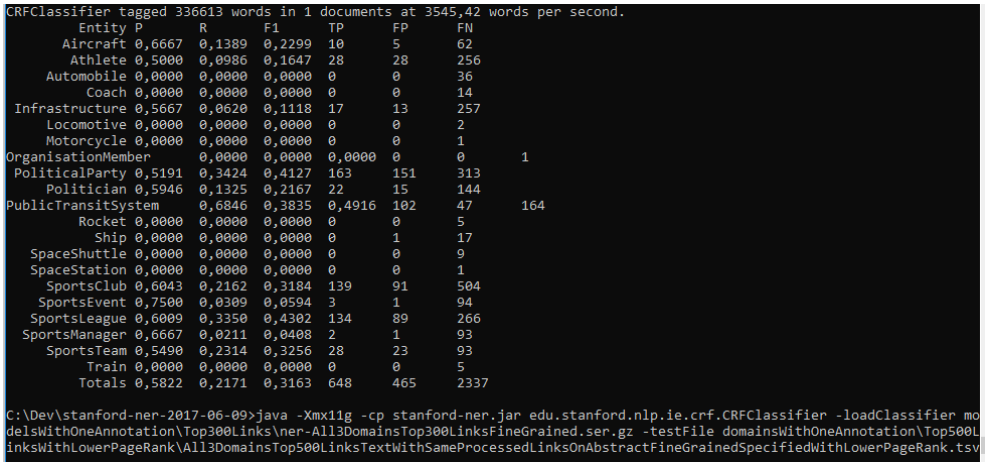


Figure 3.14: All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 500 Links With Lower PageRank

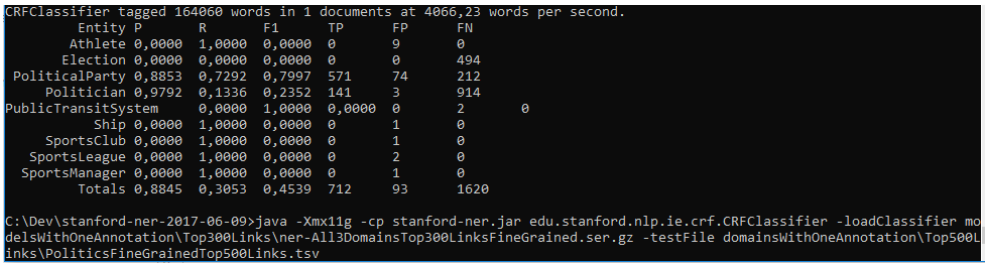


Figure 3.15: All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 500 Links

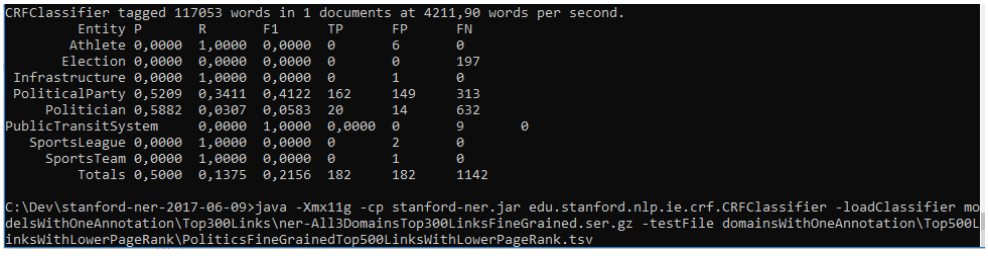


Figure 3.16: All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 500 Links With Lower PageRank

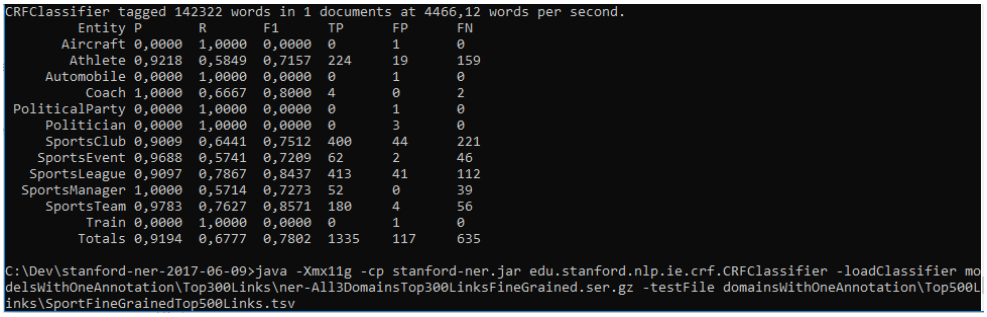


Figure 3.17: All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 500 Links

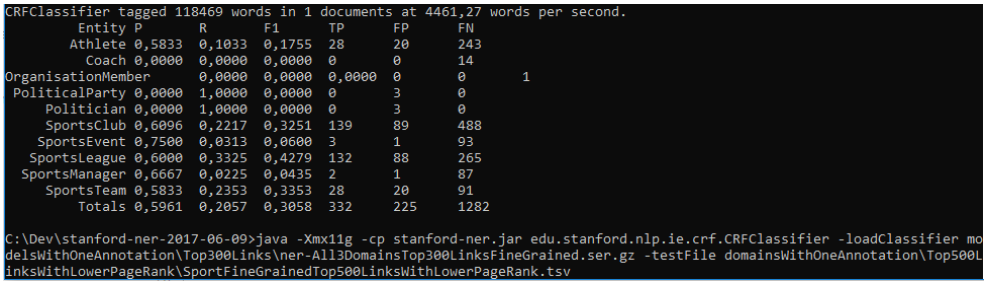


Figure 3.18: All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 500 Links With Lower PageRank

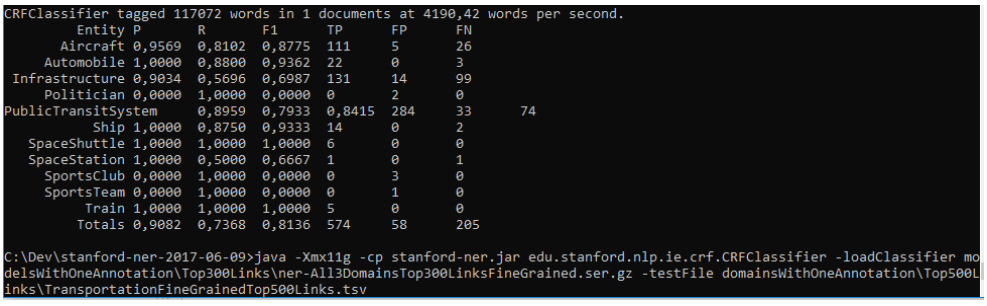


Figure 3.19: All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 500 Links

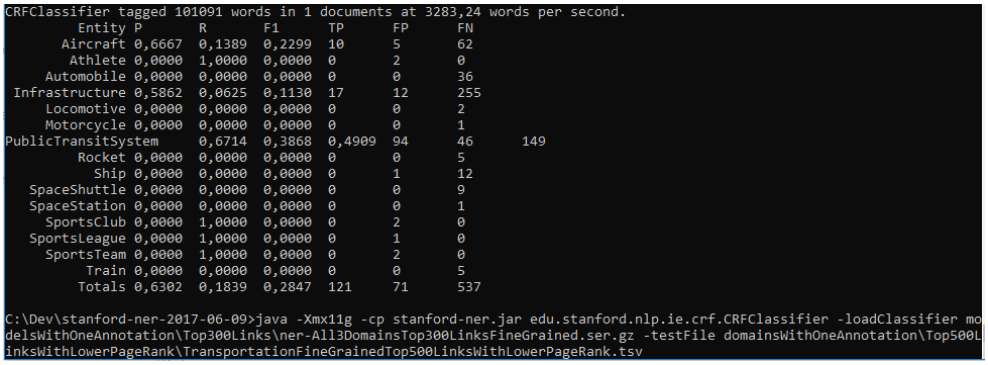


Figure 3.20: All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 500 Links With Lower PageRank

3.2.6 Experiments with higher training abstracts on model, but lower abstracts on test file

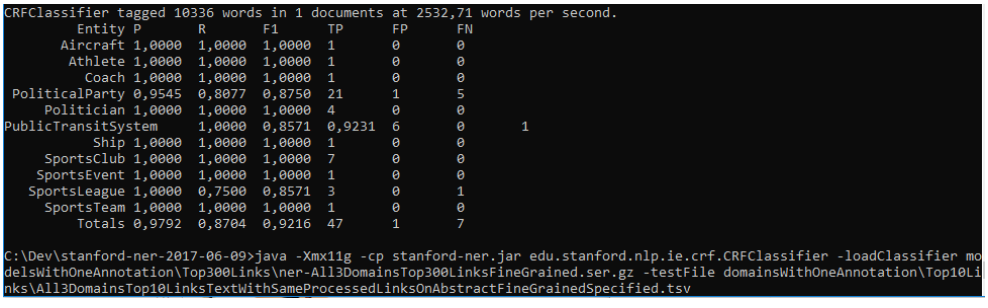


Figure 3.21: All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 10 Links

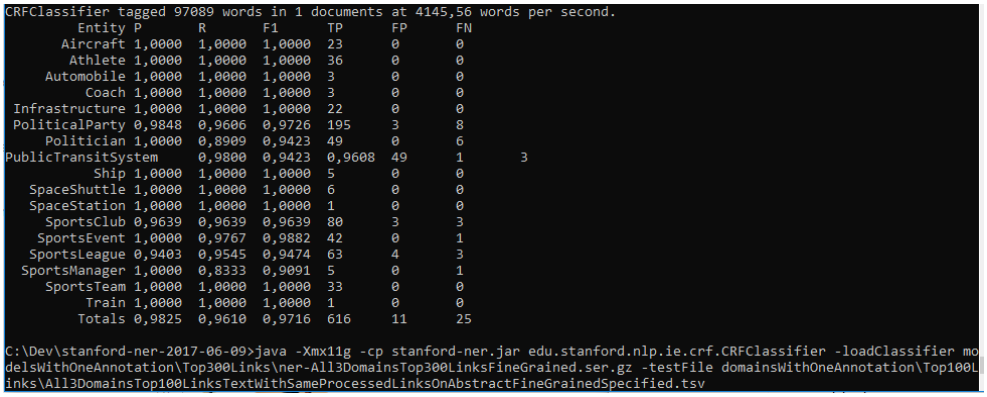


Figure 3.22: All 3 Domains Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links

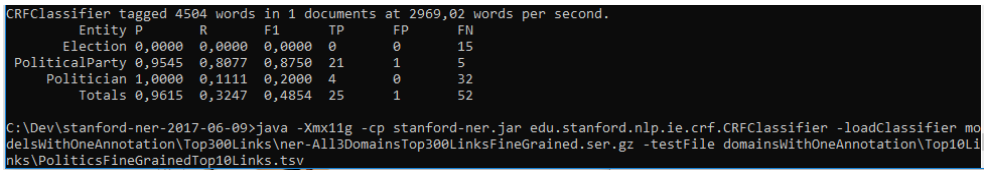


Figure 3.23: All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 10 Links

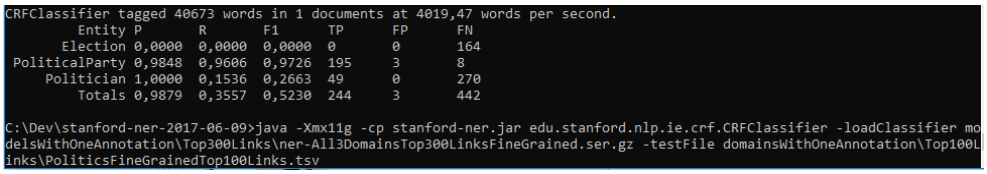


Figure 3.24: All 3 Domains Fine Grained Top 300 Links Runned With Politics Fine Grained Top 100 Links

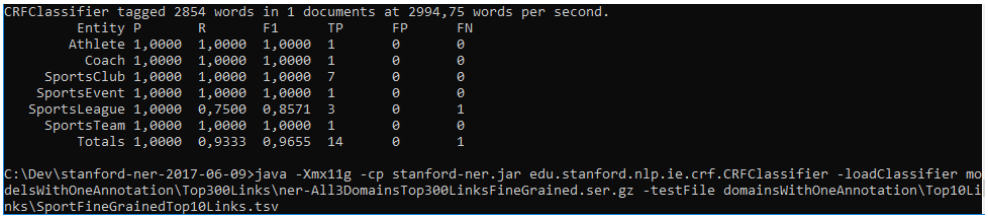


Figure 3.25: All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 10 Links

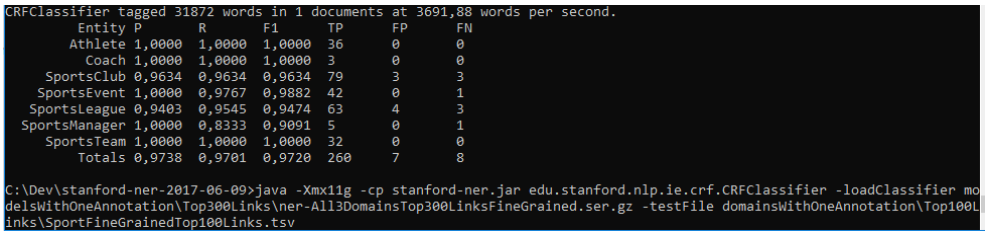


Figure 3.26: All 3 Domains Fine Grained Top 300 Links Runned With Sport Fine Grained Top 100 Links

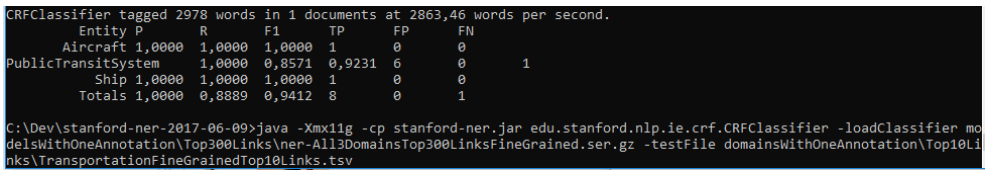


Figure 3.27: All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 10 Links

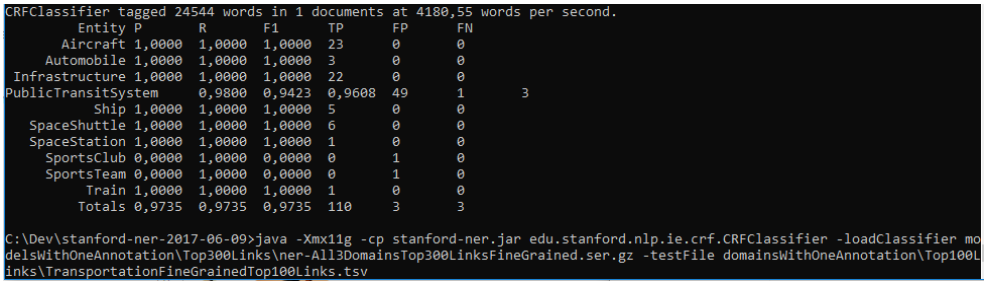


Figure 3.28: All 3 Domains Fine Grained Top 300 Links Runned With Transportation Fine Grained Top 100 Links

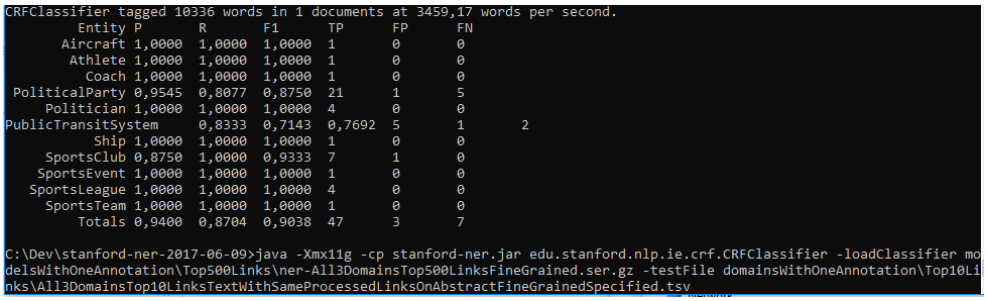


Figure 3.29: All 3 Domains Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 10 Links

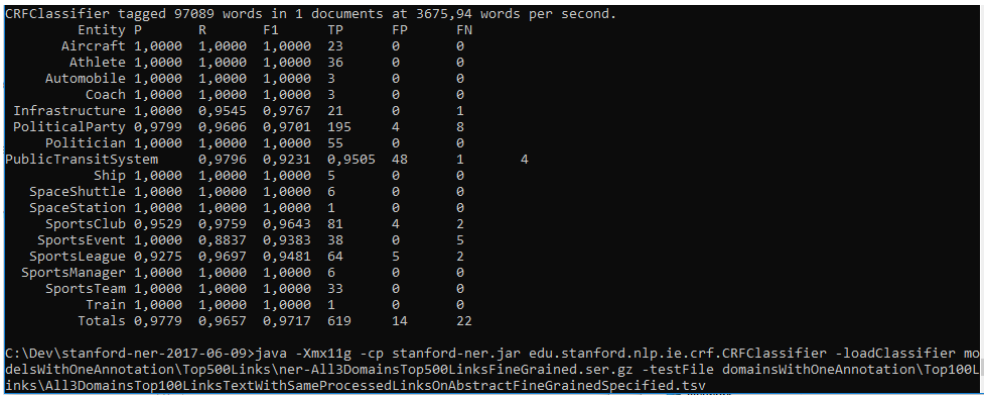


Figure 3.30: All 3 Domains Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links

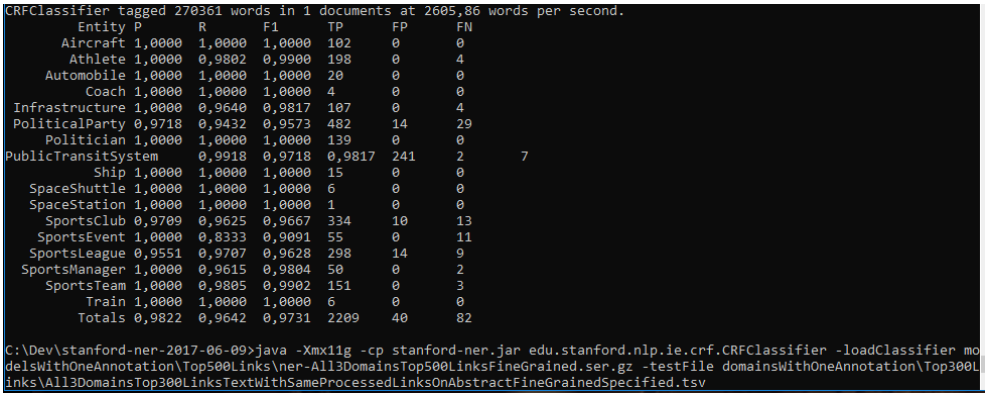


Figure 3.31: All 3 Domains Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links

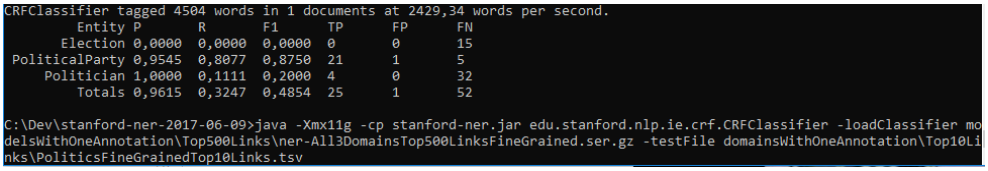


Figure 3.32: All 3 Domains Fine Grained Top 500 Links Runned With Politics Fine Grained Top 10 Links

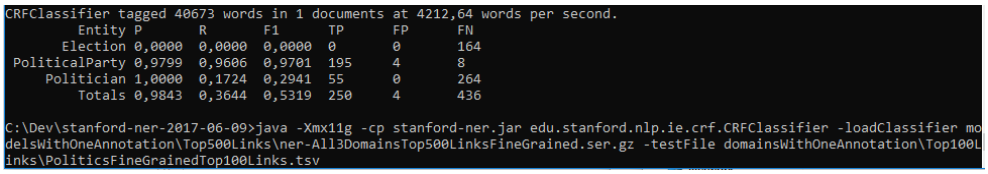


Figure 3.33: All 3 Domains Fine Grained Top 500 Links Runned With Politics Fine Grained Top 100 Links

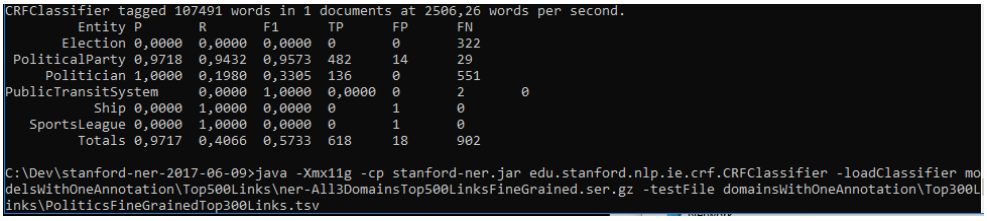


Figure 3.34: All 3 Domains Fine Grained Top 500 Links Runned With Politics Fine Grained Top 300 Links

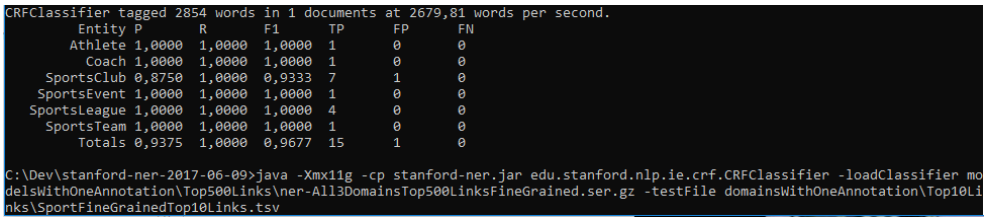


Figure 3.35: All 3 Domains Fine Grained Top 500 Links Runned With Sport Fine Grained Top 10 Links

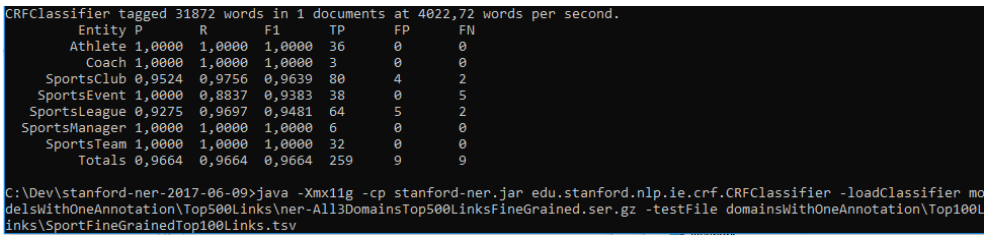


Figure 3.36: All 3 Domains Fine Grained Top 500 Links Runned With Sport Fine Grained Top 100 Links

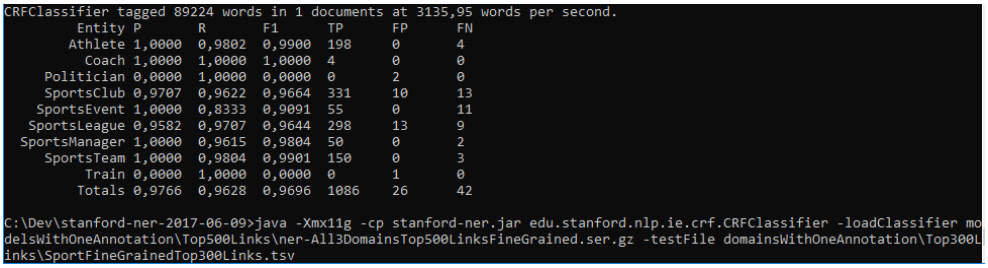


Figure 3.37: All 3 Domains Fine Grained Top 500 Links Runned With Sport Fine Grained Top 300 Links

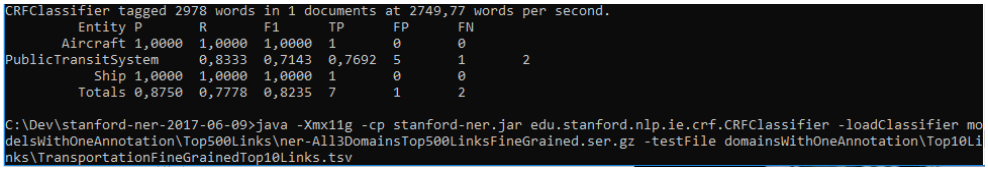


Figure 3.38: All 3 Domains Fine Grained Top 500 Links Runned With Transportation Fine Grained Top 10 Links

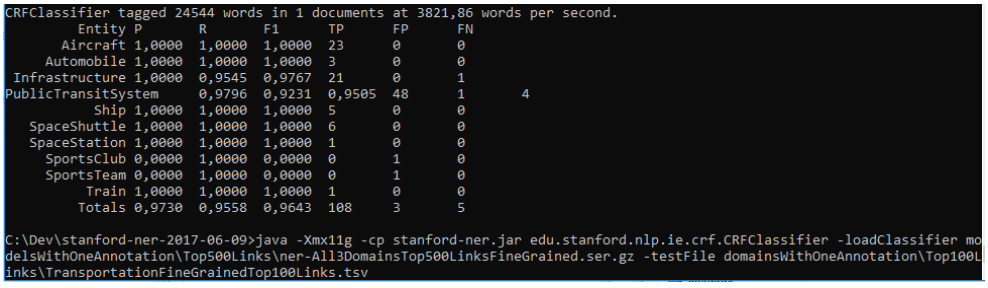


Figure 3.39: All 3 Domains Fine Grained Top 500 Links Runned With Transportation Fine Grained Top 100 Links

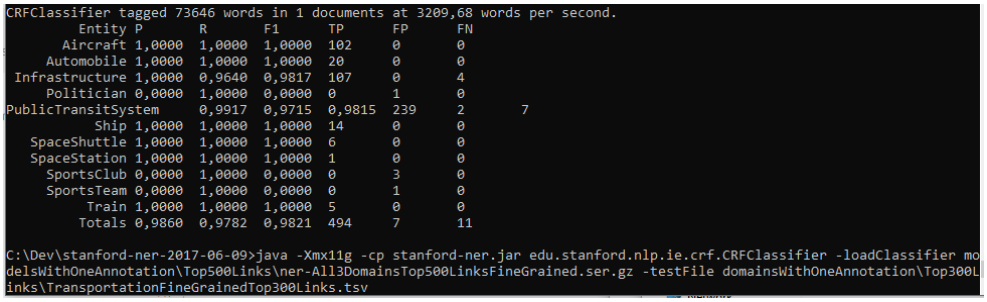


Figure 3.40: All 3 Domains Fine Grained Top 500 Links Runned With Transportation Fine Grained Top 300 Links

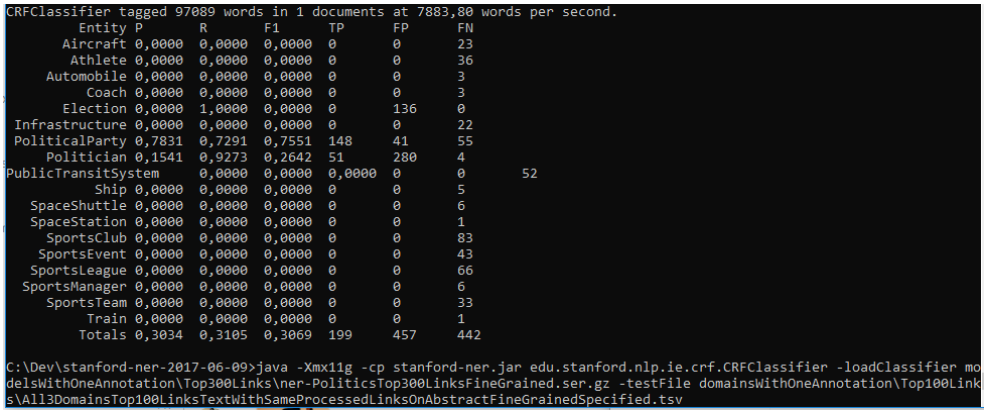


Figure 3.41: Politics Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links

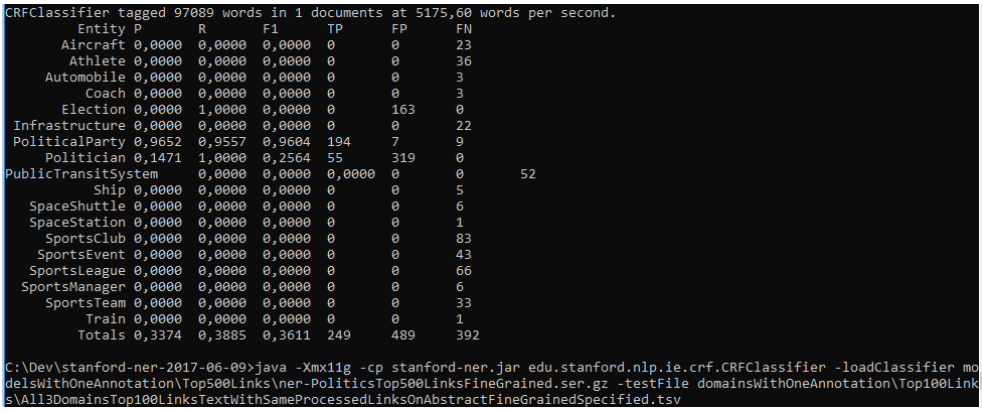


Figure 3.42: Politics Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links

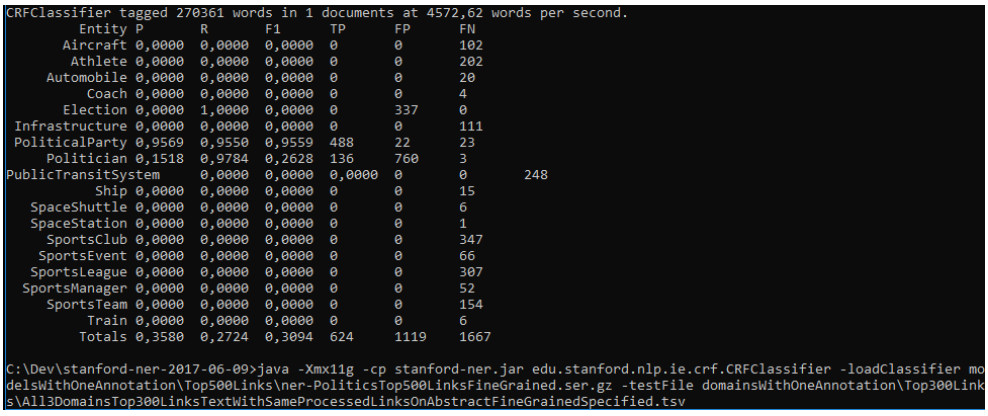


Figure 3.43: Politics Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links

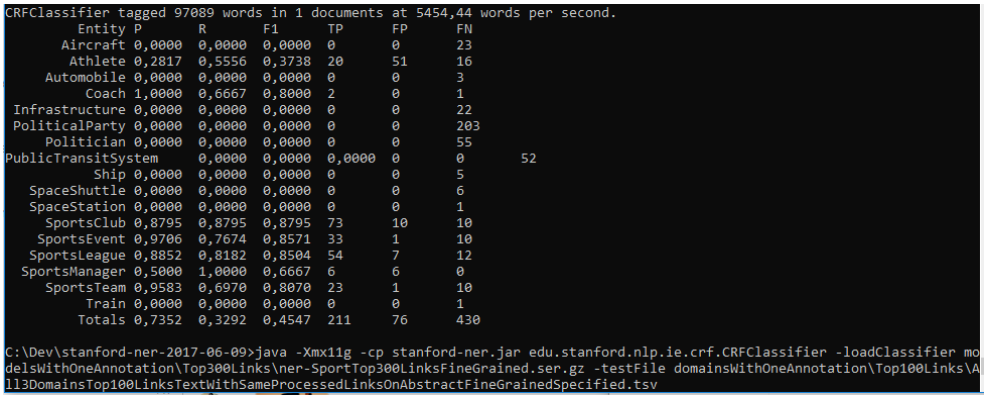


Figure 3.44: Sport Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links

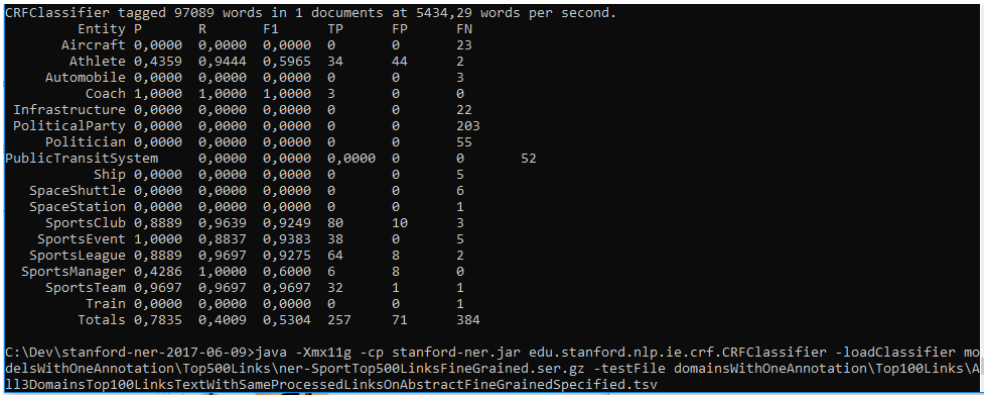


Figure 3.45: Sport Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links

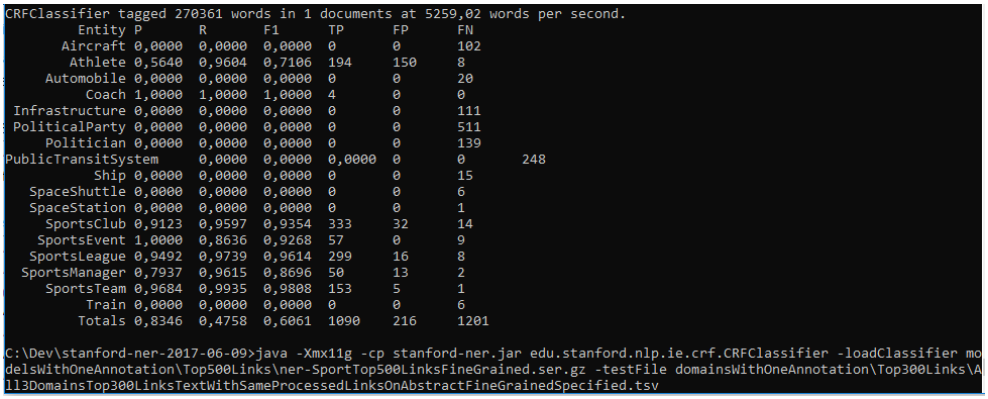


Figure 3.46: Sport Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links

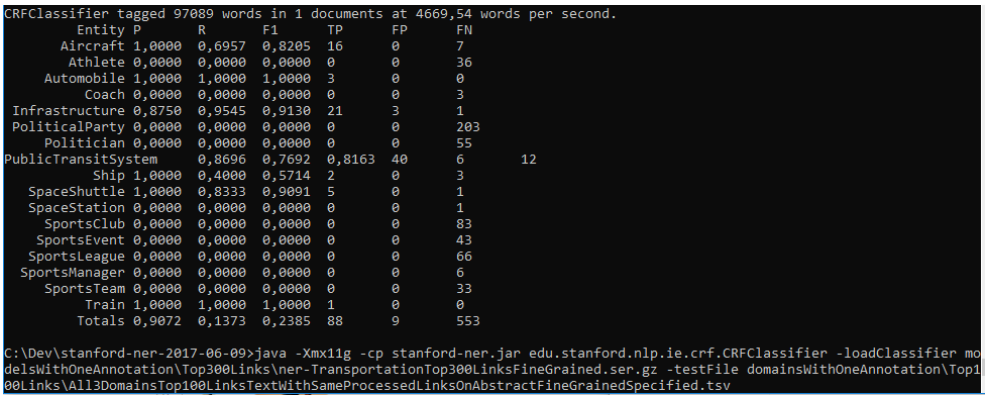


Figure 3.47: Transportation Fine Grained Top 300 Links Runned With All 3 Domains Fine Grained Top 100 Links

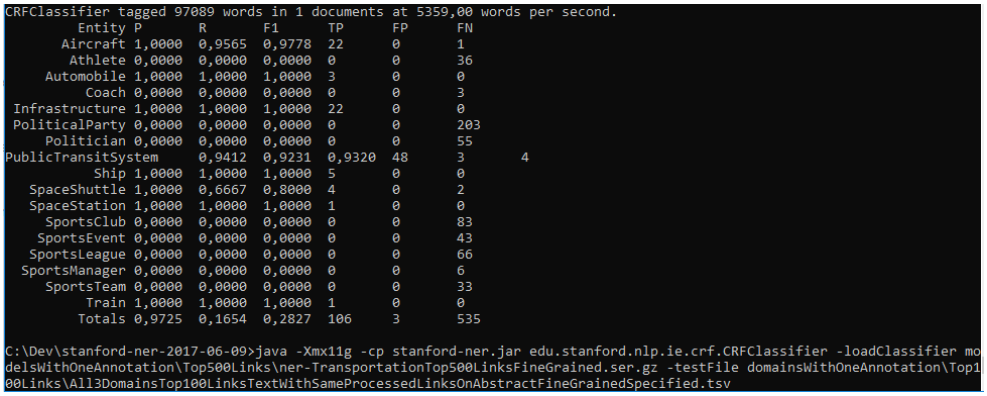


Figure 3.48: Transportation Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 100 Links

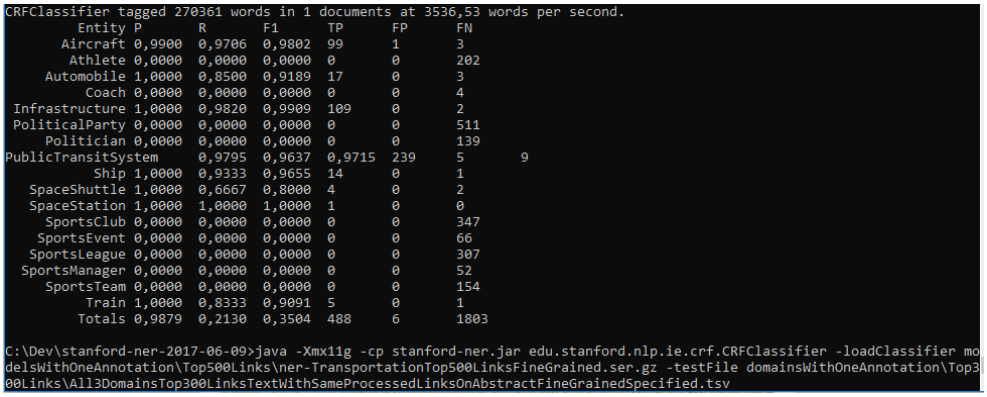


Figure 3.49: Transportation Fine Grained Top 500 Links Runned With All 3 Domains Fine Grained Top 300 Links

3.3 Summary of results

3.3.1 Graphs

Conclusion

Bibliography

[1] Information extraction. Information extraction IE. Available from: https://en.wikipedia.org/wiki/Information_extraction

[2] PageRank. PageRank. Available from: https://en.wikipedia.org/wiki/PageRank

[3] Apache Jena. Apache Jena. Available from: https://jena.apache.org/

Appendix A

Contents of CD