1 Module 1 Linguistic Fundamentals

For some sentences, I used "we" as subject. But actually only me finish this assignment. The "we" is just used for convenient!!!

1.1 Linguistic Concepts

Table 1:

Concept	Definition	Examples	NLP relevance	Reading
morpheme	morpheme is "the smallest meaningful units of language"(Bender, 2013, p.11).	*Apples* consists of apple+s, a root and plural marker.	morpheme can be used to identify the roots of words to reduce ambiguity and also for part-of-speech.	B[13] 7,12
lemma	Lemma is a basic form or a root of words and also can be explained like a dictionary form of a word.(Parshanth,[8])	Basic form of *car, cars, car's, cars'* is car. The root of *girl, girls, girl's, girls'* is girl.	Lemma can be used into Lemmatisation to get the root of word and it also can support POS-tagging and sentiment analysis.	E[18].4.2
pos-tag	pos-tag is a special label for each word in the text and this label usually means part-of-speech.	*I like hamburgers.* I and hamburgers are nouns, like is verb. And in other context, *Sky like a carpet.* Sky and carpet are nouns, a is an article and like is a preposition.	pos-tag can be used into pos-tagging, which can be used to enhance word-based features.(eg: book my hotel, I like this book. Two book are different part-of-speech). And for sentiment analysis, it is also useful especially for lexicon based method. And it is also useful for Named Entity Recognition.	E[18] 8.1; JM[26] 8.1, 8.2
constituent	Constituent is a single word or phrase in a sentence and its function likes a single unit within a hierarchical structure. And most constituents are phrases. (Wiki,[4])	Tom eats [a very delicious pancake]. When we analyze constituents, the *a very delicious pancake* are noun phrase and it is a constituent in this sentence. And in sentence *Sky like a blue carpet*, *a blue carpet* is noun phrase and it is a constituent.	Constituent can be used to analyze composition and structure of phrases in sentence, which can be helpful to do analyze the relation between phrases to finish certain task such as getting sequence of sentence, modality and so on.	B[13] 51, JM[26] 12.1 - 12.3
dependency	Dependency means there is exactly a node in the sentence structure which corresponds one elements(word, morph).(NLP,[5])	For example, in sentence *Monkey likes eating bananas*. The roots is *likes*. And dependency tree: *Monkeys* $\leftarrow$ *likes* $\rightarrow$ *eating* $\rightarrow$ *bananas*. *likes, monkeys* is nsubj relation. Monkeys is subject and depends on likes.(NLP,[5])	Dependency can be applied into extracting opinion and sentiment analysis. Because dependency can show subject's direct attitude. And dependency structure can illustrate root node, head of structure and syntactic analysis.	JM[26] 15.1, 15.2

1.2 Linguistic Analysis

1.2.1 UD POS-tag

1. House Speaker Nancy Pelosi, D-Calif., also knocked the president's funding request, calling it "completely inadequate" and criticizing the president for previous moves to cut funding to public health programs.

Words	UD tag	Words	UD tag	Words	UD tag	Words	UD tag
House	PROPN	Speaker	PROPN	Nancy	PROPN	Pelosi	PROPN
,	PUNCT	D-Calif.	PROPN	,	PUNCT	also	ADV
president	NOUN	's	PART	funding	NOUN	request	NOUN
,	PUNCT	calling	VERB	it	PRON	"	PUNCT
completely	ADV	inadequate	ADJ	"	PUNCT	and	CONJ
criticizing	VERB	the	DET	president	NOUN	for	ADP
previous	ADJ	moves	NOUN	to	PRT	cut	VERB
funding	NOUN	to	PRT	public	ADJ	health	NOUN
programs	NOUN	,	PUNCT				

Table 2: UD POS-tag for first sentence

2. She called the response "anemic."

Words	UD tag	Words	UD tag
She	PRON	called	VERB
the	DET	response	NOUN
"	PUNCT	anemic.	NOUN
"	PUNCT		

Table 3: UD POS-tag for second sentence

1.2.2 Morphological analysis

Morphological analysis:

Funding has two part-of-speech: Noun and Verb. Here are both noun and it is combined by fund(root) and inflectional suffix ing(tense marker). And we can see in this sentence it used funding not fund. Because fund means money but funding means the act of collecting the money. And here the sentence emphasizes the act.

Completely is an adverb and it is combined by complete(root) and inflectional suffix ly. It means finished. The *ly* transforms the pos-tag from adjective into adverb, which means it emphasizes a degree. Thus, the *ly* means this word is adverb.

Inadequate is an adjective and it is combined by adequate(root) and in. It means insufficient. The pos-tage does not change when adding in but when adding *in* the meaning become opposite: from sufficient into insufficient. Thus, the *in* expresses a negative meaning.

Criticizing is a verb and it is combined by criticize(root) and inflectional suffix ing(tense marker). The *ing* transforms criticize into criticizing and *ing* give the information that the verb happens right now or comes along with action of subject.

1.2.3

Funding is quite ambiguous in some sentences. Because funding have two different part-of-speech. 1. For example: He cut the funding. In this sentence funding is noun and means the action of collecting money. 2. Another example: Who is funding this project?. This funding is verb and is combined by fund(verb)+ing(tense marker). And *ing* means the verb happens right now or come along with subject.

1.2.4

The virulent germ we now call the Spanish flu happened to strike at a diabolical moment in the history of politics and propaganda.

The obvious mistakes is the nsubj line from call to germ should be from happened to germ. And The line(xcomp) pointed from happened to strike should also be added.

2 NLP tasks: definitions and identifying features

2.1 NLP Tasks: Gold Data

2.1.1

2.1.2 BIO-style

The BIO-style:

(a) Protesters went to the hague with banners “Rutte Will Let Us Drown” based on predictions on future flood risks. O O O O B-LOC O O O B-PERSON O O O O O O O O O O O O O

(b) The Institute for Environmental Studies (IVM) in Amsterdam-Zuid uses the Aqueduct Floods web tool which is developed for determining Flood Risks globally.

O B-ORG I-ORG I-ORG I-ORG O B-ORG O O B-LOC O B-LOC O O B-ORG I-ORG O O O O O O O O O O

(c) The results of the researchers led by VU water and Climate Risk Experts Timothy Tiggeloven and Philip Ward have been published in the journal Natural Hazards of Earth Systems Sciences.

O O O O O O O B-ORG B-PERSON I-PERSON O O O O O B-PERSON I-PERSON O B-PERSON I-PERSON O O O O O O B-ORG I-ORG I-ORG I-ORG I-ORG I-ORG

2.1.3 Allen NLP system

First: There are several mistakes at first sentence. hague in Allen is not a location name. This mistake is caused by capitalization. And another mistake is that Allen markes *Rutte Will Us Drown* as organization. The possible reason is these four words use capitalization.

Figure 1: First sentenc result of Allen NLP system

After correction it matches my answer:

Figure 2: First sentenc result of Allen NLP system

Second: There are two mistakes: Zuid should be LOC, Flood Risks should be nothing. For mistakes of Flood Risks it is because of capitalization. And for mistakes of Zuid, I think this is because of the - and the sequence of word. And when I change into *Zuid, Amsterdam*. The model show right answer.

Figure 3: Second sentenc result of Allen NLP system

After correction it matches my answer:

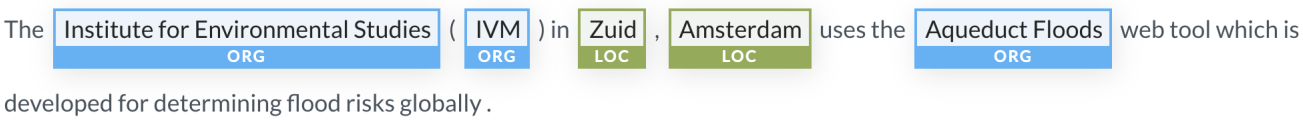

Figure 4: Second sentenc result of Allen NLP system

Third: There are one mistakes: Climate Risk Experts should be nothing. This is because of capitalization.

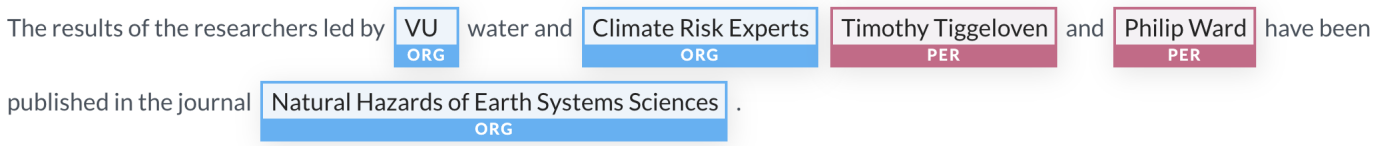

Figure 5: Third sentence result of Allen NLP system

And after correction it matches my answer.

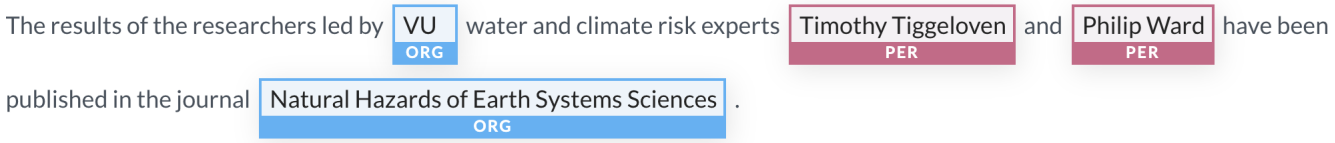

Figure 6: Third sentence result of Allen NLP system

2.2 NLP Tasks: Features

2.2.1 Sentiment Analysis

For sentiment analysis, we focus on machine learning method.

Features: Feature of single words:1. Tokenization(Just split words by space) 2. Capitalization 3. Symbol or Emoji. 4. Negations Handling and Expanding contraction(find symbol like ' and also can detect some words like *n't, 'll,'d*) into *not, will, would, had* 5. POS-Tagging.

Design choices: 1. Firstly tokenization is used to split the sentence or paragraphs into every single words by white space. 2. Second we can extract capitalization as feature. Because capital letter will show strong emotions. For example:*SO GOOD!*. 3.Another feature is also Symbol or Emoji, which can imply the emotion such as :) , !!!!!!. Thus, we can have a function to check the symbol or emoji in sentence. 4. Furthermore, negations is vital feature because system can not detect *n't* as *not*. Thus, we can have a function to handle the negations. 5. Furthermore, pos-tag is also important feature,w which is for obtaining the pos-tags as features. Because we can mainly focus on adjective as feature to train the model or we can use all words.

This part the information is mainly form [20]

2.2.2 Part-of-speech tagging

For Part-of-speech tagging, lexicon method is mainly discussed in this part. Lexicon based method is basically using a dictionary to match each word to get certain attribute. For example, for lemma lexicon we match apples with apple in lexicon to find lemma. Features: Feature of words: 1.Expanding contraction 2. Tokenization, we know this is Noun. 3. Capitalization 4.Lemmatization. 5. Morphological analysis - morphemes. For example for some words end with *tion* are noun. 6. Previous token. 7. Following token ([25],114)

Design choices: 1. For POS-tagging, the most important is feature of words. First all the contractions should be expanded. For example, *don't-do not*. So we need a function to detect the contraction and expand it. 2. Then next step is Tokenization to get every single word. 3. Then we use capitalization as feature. For example the first letter of the words or all words are capitalised, which are usually noun. Then we can have a preprocess to check whether the first letter is upper case. 4. Lemmatization and Morphological analysis are also useful for this task. For example *dangerous*, we first have a lemma lexicon to get word lemma *danger*. Then we use morphological analysis to analyze the word end with *ous*, this is feature of adjective(also a lexicon). 5. For some words have multiple pos-tag. We can have a function to check previous token or following token. For example *links have two part-of-speech (Noun),(Verb)*. If it is noun, the fallowing words are usually not noun. *There links are useful*. If it is verb, it usually followed by noun. *Here links Beijing*.

2.2.3 Named Entity Recognition

For Named Entity Recognition:

Features: Feature of words, 1. Tokenization 2.POS-tagging 3.Capitalization 4. Previous token. 5 Following token. 6. Gazetteer (from assignment feedback)

Design choice: 1. For name entity recognition, first feature is Tokenization. The computer basically split the text according to the space. 2.Then the next feature is POS-tag. POS-tag is important for named entity. Because most named entities' are noun. 3. Furthermore, capitalization feature is also useful. This feature can be described like low case: *up, forecast*, full cap:*USA*, first cap:*China, Wuhan* and so on.[1]. We can have a function to get these capitalization. 4. We can have a function to check previous and following token. For example, if previous token is *Dr.* we know the following token may be a name. 5. Gazetteer is a lexicon which include the know name. Thus, we can have a function to match these name in text. This part the information is mainly form [23].

3 Crosslinguistic variation

3.1 Linguistic concepts across languages

3.1.1

Classification of writing system is quite diverse, and the broad categories are alphabetic writing system, syllabic writing system, and logographic writing system. Writing system can influence the tokenization. For example, alphabetic writing system-English split the word by space. While for logographic writing system-Chinese, tokenization is tricky work because there is not space between words. And there are lexicon based or machine learning algorithm such as LSTM-CRF methods can be used for tokenization. ([18],186)

3.1.2

For English, plural marker: apple*s*,orange*s*. Agreement marker: she sing*s*. *s* means subject is first person plural.

For Portuguese adjectives there are gender maker: male-end up with **o**: louc**o**. female-end up with *a*:louc**a**. [16]

Eastern Pomo has evidential maker to describe the evidence of the message. puna kayan-*a!*(I saw the fire in person). puna kayan-*e!*(I didn't see it, but I feel there was a fire). puna kayan-*ye!*(I didn't see it, but I heard there was a fire)puna kayan-pek!(I didn't see it, but there is a evidence can guess a fire happened. [6]

3.1.3

Morphologically rich means expressing different meaning or grammatical relations can be achieved by changing words. In other words a sea of noun forms or verb forms can be derived from only one word root.

For example: Hebrew. About 120 inflection of each verb-root + prefixes and suffixes. Each noun has about 75 forms. For example each noun have gender morpheme, number morphemes, tense morphemes and so on to end with.[17], [7]

3.1.4

In most case, the main difference of inflectional and derivational morphology is if it will change part-of-speech or meaning of words. Inflectional morphology never change part-of-speech of words. For example, +*est*: small(adj) and smallest(adj). While for derivational morphology it usually changes part-of-speech or meaning of words for example, part-of-speech change: +*er*: sing(verb) and singer(noun). Meaning change: +*un*: important(adj) and unimportant(adj).

For Chinese, inflectional morphology can be achieved by plus *men* to describe plural format(which is similar to +s/es in English),+*men*: $w\check{o}$(I, Personal Pronoun), $w\check{o}men$(us , Personal Pronoun). *ta* (he, Personal Pronoun), *tamen*(they , Personal Pronoun). For derivational morphology adding *de* can change adjective into adverb(which is similar to +ly in English). For example: $Qingyi$(easy, adjective), $Qingyide$(easily, adverb). noun).[12]

3.1.5

Agreement is the form of one word's changes dependently on the form of another word or phrase in a sentence. For example, *I am, he is*. And there is no *I is, he am*. And also *this/that apple* not *those/these apple*. [9],[13]

3.1.6

Mechanisms:

1. Word order. For example: *John hates Mike* and *Mike hates John*. They are *normal case* and obey *subject-verb-object*.

2.Agreement. There are some agreements in the sentence which correspond a certain grammatical function. For example: *Zuek lagun-ei opari polit-ak ema-ten dizkiezue.* means *You always give nice presents to your friends*. *dizkiezue* is auxiliary corresponding presents and marked third person plural absolutive argument. *-ei* is third person plural dative argument, which corresponds *friends*. And *-ten* is second person plural ergative argument corresponds *Zuek, you*. Thus in this way, we can see each grammatical function. (example from[13], page 94)

3.Case marking. Case means that the noun phrase changes depending on the role in sentence. For example, adposition in Japanese, *Jay **ga** Kay **ni** a book **o** gave*(This is a Japanese-English combination example). **ga** indicates Jay is subject of verb *gave*. And **ni** indicates Kay is indirect object and **o** indicates a book is direct object. (example in[2],page 2)

3.1.7

For non-projective structures language word order is not useful. Word order is suitable as a feature for English in NLP because we can get many information from word order to finish certain task. For example, *I missed the flight this morning*. We can easily get information of the subject, direct object and so on from word order to perform some NLP tasks such as POS-tag.

But for non-projective structures. For example, in English *It is what country support should try to achieve*. We can see the word order is quite free and it is hard to get the any information simply from the word order. In this case dependency three is more useful. [15]

3.2 Dealing with different languages

The fundamental difference between Chinese and English is that Chinese is logographic system language and each character represents itself not combined by letters. And also Chinese does not need space between two different words. [3]

Tokenization is a trick task for Chinese because Chinese is different English, and there is no space between the characters. And a Chinese word can be composed by one or more Chinese characters, which means its meaning can be interpreted in terms of its composite characters. Thus, morphological analysis and syntactic analysis should be used into Tokenization. Morphological analysis can be based on compounding, affixation, and conversion. Compounding is quite common in Chinese and has various categories. For example,*CaiFu(wealth,Noun) = Cai(money,Noun)+Fu(wealth,Noun)*. *XinTeng(care, Verb) = Xin(heart,Noun)+Teng(painful,Adj))*. We can see if we don't have morphological analysis and just simply split every character, which will bring lot of error on POS-tag or sentiment analysis. Chinese is quite similar to English in some syntactic analysis. Chinese also obey the SVO-Subject Verb Object. For example, *Tamen Zai Kan Dianshi* means *They are watching TV*, one to one corresponding.[21]

For sentiment analysis, 1. tokenization 2. Pronunciation and Intonation 3. POS-Tagging 4. Symbols and Emoji

1. first step is Chinese tokenization. Here we use lexicon based method. The main idea is first used long combination of characters to match the lexicon. If it doesn't match, we reduce one characters at the end of combination and continue the match. For example, *Ta(She) Shi(is) Ge(a) Hao Ren(beauty)*. First we match long combination *Tashigehaoren* and we find there is no match. Thus, we reduce *ren* and become *Tashigehao*. Then we repeat till we find *Ta* matches the lexicon. 2. Next feature are pronunciation and Intonation. A same *pinyin(Phonetic symbol in English)* has four intonation in Chinese. And each intonation has various words. For example: $(h\check{a}o, good), (h\bar{a}o, pull), (h\acute{a}o, scream), (h\grave{a}o, like)$. We can see when intonation are $\check{a}$ and $\grave{a}$ they show positive sentiment. Thus, intonation is good as a feature for sentiment analysis. This also can get from lexicon after tokenization. 3. Then for POS-tagging using lexicon to determine the part-of-speech. 4. Again we can have a function to symbols and emoji. Furthermore, since we used lexicon based method we also need a lexicon to store each token's score of sentiment. Then for one sentence, text or paragraph we can calculate total score to get sentiment.[21]

For part-of-speech tagging, 1.tokenization 2.N-grams.

1.The difficulty of POS tagging is still tokenization. Because as we discuss before the way of split the words directly change the part-of-speech. And here we still use lexicon based method. And the method is the same as sentiment analysis and after tokenization we match each word in lexicon to determine the part-of-speech. 2. Another feature can be used is N-grams. Here we use 2-grams, which means we only need to check one more characters before or after current character. For example, *Ta(She) Shi(is) Ge(a) Hao Ren(beauty)*. First we check *Ta* since it is 2-grams we check *Tashi*. And we found there is no match. Then we just leave *Ta* alone. Next, we check *shi*, which means we need to evaluate *Tashi* and *Shige*[24]. Then, after splitting sentence into words we match each word in lexicon to determine the part-of-speech.

For named entity recognition, 1. tokenization. 2.N-grams 3. POS-tag. 4. previous and following token. 1. Tokenization is still important. But for NER we think we should split the whole sentence into every single characters. 2. Next we apply 2-grams, 3-grams or even more to combine every single characters into words.

Words is important features because NER of Chinese are usually words not characters. 3. Then, POS-tag is also a vital features because NER is usually are Noun. With these features, we can use machine learning method to predict NER. 4. Previous and following token are also useful for Chinese NER. For example *Boshi(Dr.)* is usually followed by a name.

4 NLP tasks and Gold Data

4.1 Gold data and Evaluation

4.1.1 Creating a Gold Standard

Inter-annotator agreement of two annotation was calculated in this part, which is measured by the Cohen's Kappa which can determine the accuracy of classifier. The main idea is using confusion matrix(7) to calculate the Cohen's Kappa. And the result is calculated by equation 1

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

where $p_0 = \frac{FF+NN+AA}{total}$ is same prediction of two annotators, $p_e = \frac{a_1b_1+a_2b_2+a_3b_3}{total^2}$, And a_n, b_n are sum of numbers for each label.

And the result of two annotations is 0.222, which has low agreement accuracy. We all know the famous quote *There are a thousand Hamlets in a thousand people's eyes*. Thus,towards tweet different people certainly have different opinions. And these two annotations were made by student, and person's evaluation is subjective.

Annotation 1 \ Annotation 2	F	N	A
F	5	2	4
N	1	3	3
A	2	1	4

Figure 7: Confusion matrix of two annotations

4.1.2 small experiment

In this part we did small sentiment analysis of movie review. The 2000 reviews are from Pang and Lee movie review data. First we split the 2000 reviews into 1600 train set and 400 test set. We use two different method to compare the accuracy and F1 measures.

First, lexicon method has been used. In this part we used SentiWordNet lexicon, which contains part-of-speech, negative scores and positive scores for each words. We firstly expand contraction of review, for example expanding *don't* into *do not*. Then using python function to do the tokenization. After tokenization each words match the words in lexicon to calculate the negative and positive scores. And if score is below 0 the review is negative otherwise it is positive.

Secondly, we use machine learning method. The words vector of train set was created first, and then we used Logistic Regression to train the words vector with corresponding label and created a classifier . Final step is using classifier to predict labels of test set.

The accuracy and f1 measure of lexicon based classifier are 0.69, 0.58. And accuracy and f1 measure of logistic regression are 0.81, 0.81. And firstly the a two-tailed binomial hypothesis test was used to measures the difference in accuracy. Using lexicon based method as base line, the p-value is $1.83 * 10^{-152}$. And we also used proportions , the p-value is also super low, which is $2.6 * 10^{-6}$. It means the accuracy of two classifier have significant statistical difference. That's because lexicon based classifier's result greatly depends on preprocess of data. Some process such as remove stop-words did not include in the lexicon based classifier. Thus, the lexicon based classifier has low accuracy when compared with logistic regression.

5 Module 2 Coreference Resolution

For some sentences, I used ”we” as subject. But actually only me finish this assignment. The ”we” is just used for convenient!!!

5.1 Understanding the task and representation

5.1.1

5.1.2

5.1.3

The coreference chains in the text:

Chinese tourist: [Chinese tourist, A Chinese tourist, she, the 61-year-old woman, She, her, she, the woman, The ill Chinese woman, Her, The Chinese woman, Her],

Public Health Minister Anutin Charnvirakul: [Public Health Minister Anutin Charnvirakul, Mr Anutin, Mr Anutin, he, Mr Anutin, Mr Anutin, Mr Anutin, The Public Health Ministry,he],

59 people in China: [59 people in China, them, All, They],

Health officials:[Health officials, They]

The new strain of coronavirus: [the new strain of coronavirus, the new strain of the coronavirus, which, the virus, the virus, the new coronavirus,it, the new virus strain]

5.1.4

5.1.5

Singleton: Thailand, hospital, Monday, Bamrasnaradura Infectious Diseases Institute, Nonthaburi province, fever, respiratory symptoms, doctors, clearance, home, flight, outbreak of pneumonia, central China, markets, animals, seafood , Wuhan city, workers, buyers, transmission.

Table 4:

Real-world entity	Mentions	Singleton?	Anaphoras	cataphoras	bridging anaphora
A Chinese tourist	1.A Chinese tourist was found 2.the 61-year-old woman was 3.close to the woman 4.The ill Chinese woman.	No	1. when she arrived in Thailand 2. She now had no fever. 3.gave her a clearance she would be		
the new strain of coronavirus	1.with the new strain of coronavirus. 2. infected with the new strain of the coronavirus. 3. transmission of the virus. 4.with the virus outside	No	1.**which** has been linked to		
China	1. people in China 2.outside China	No			
Public Health Minister Anutin Charnvirakul	1.Public Health Minister Anutin Charnvirakul 2. Mr Anutin 3. said Mr Anutin. 4.Mr Anutin said.	No	1.he said		
Sixteen other people who	1.Sixteen other people who	N0			1.**the results** were
59 people	1. 59 people	No	1.One of them died 2.All had attended big markets 3.They were either workers		

5.2 Towards designing a system

5.2.1 Analyzing the phenomenon

1)

Table 5:

Item	Antecedent	Reason
Mr Anutin	Public Health Minister Anutin Charnivirakul.	1. Match name Anutin 2. Mr is man and there is only male in sentence(From assignment 2)
She	61-year-old woman	1. She means female and there is only female in sentence. 2. Woman is close to she.
them	59 people	1. Them means many people and 59 people is closer to this word them than sixteen other people. Thus, we choose 59 people.

2)

1. Kim and his brother bet that Kim could take a picture of himself while doing a handstand.

Antecedent: Kim. According to the syntactic rule this himself can only represent Kim.

2. Kim and his brother bet that Kim could take a picture of him while doing a handstand.

Antecedent: his brother. Because if antecedent is Kim here can only use himself. So if it is him, it represents his brother.

3.I went with John to the market yesterday and then took a walk in the forest with Mark. He was in a cheerful mood.

Antecedent: Mark. Here we can evaluate the structure of sequence. If he means John, the second sentence should follow *yesterday*. Now it follows *Mark*. Thus, the antecedent is Mark

4. John met Mark for dinner. He felt like having company.

Antecedent: John. We can also the structure of sequence. Subjects refer to subject. Thus John refers to He. (From assignment feedback)

5. Kim was stopped by a policeman. He was not pleased.

Antecedent: Kim. Because stopped is a passive word and it emphasizes the subject Kim. Thus, he is Kim.

6. John invited Mark for dinner. He knew he was broke. Antecedent: The second he is Mark. We can also the structure of sequence. Subjects refer to subject. Objects refer to Object. Thus, the second he is Mark. (From assignment feedback)

7. John invited Mark for dinner. He knew he could please him with his latest recipe for lava cake.

Antecedent: Second he is John. We can also the structure of sequence. Subjects refer to subject. Objects refer to Object. And in this sentence It is more logical to say first two he are parallel because there is still a him. Thus, the both two he are John.

8. John has a bike, but no car. It is red.

Antecedent: bike. We can also use structure. The last menthion is car Because there is a negation-no and it means John has no car. Thus it can only means Bike.

9. He brought flowers and chocolates. They tasted great.

Antecedent: chocolates. They means many and in this sentence it must be chocolates. And because only chocolates can use taste.

10. She poured water from the pitcher into the cup until it was full.

Antecedent: Cup. Because the water is from pitcher into cup. And full means it is cup

11. She poured water from the pitcher into the cup until it was empty

Antecedent: pitcher. Because the water is from pitcher into cup. And empty means it is pitcher

3)

Sentence a, b are Syntactically impossible. Because if Kim want to take a picture of Kim. The format should be himself, otherwise it is him.

sentence c,d,f,g,h are structure, we can use sequence of structure and one to one corresponds.

Sentence i,j,k should use word knowledge. It is more logical to interpret these sentences.

Sentence e is special, we can just use our intuition to feel it.

5.2.2 Towards system design

System

(a) For syntactic feature. First we should have a feature about defining the type of mention, like proper noun, a common noun, or a pronoun. In this way it can help to find conference. Next we can have a feature about how many letters overlap of two mentions. This feature aims to link the mention like The United States and States.

For semantic feature. Gender feature is needed. For example. Mr. is he and Miss. is she. We can also include number feature. For example, a, an, this are singular. Those, these are plural. Another feature is word before head word. We can investigate whether the words before two head words match. For example, CEO Cook and CEO Rice.

For the structure. Feature of position is necessary. We should measure the distance of two mention. For example, whether they are is same sentence, whether they have another mention in between and so on.

For world experience, it is difficult one to figure out the feature. The possible feature is presence of lexicon. For example, when the text has USA and president we can have a lexicon to match this information - Donald Trump. (From QA session 27/05.)

(b) 1. For determining type of mention, POS-tag can be used here. Then we can get proper noun, or common noun, or pronoun.[18] Then the computer can only focus on these part-of-speech. 2. And then we can have a feature to count have many letters overlap with head. like The United States and States. We count that there are six letters overlap, which means they probably are same thing.[14] 3. For semantic part, first we have gender feature. For example, if *Mr.* is in the sentence, the computer will get the information : Male, singular(From lexicon) and then we can match *he* in the text.(from assignment feedback) 4. Number feature. For example, if *a/an* is in the sentence, the computer will get the information : singular(From lexicon) and then we can match head in the text. (from assignment feedback) 5. We check previous token or following token. For example, CEO Cook and CEO. 6. Position feature. We need to check the how many tokens between the two mention and if there is a token is comma.[14] For example, . . . *Mike, he* . . . . There are only one token between *Mike* and *he* and it is comma. Thus, we can conclude he represents Mike. 7. We can create a lexicon to match the information for world experience. For example, when the text has USA and president we can have a lexicon to match this information - Donald Trump(he,singular). (From QA session 27/05.)

(c) 1. token. (whether the token has been previously been observed as being part of a name is useful information to determine whether it is currently part of a name). (From assignment feedback)

2.Pos-tag. The system can focus on proper noun, or common noun, or pronoun.

3. Gender and Number. As we discussed at section b) these feature can narrow the scope of mention.

4. Previous token and following token. If the previous token is a title (Mrs, Mr, Miss) or following token is company name or something. We can get the information about gender, number or something else.

5. Position feature. For example, . . . *Mike, he* . . . . There are only one token between *Mike* and *he* and it is comma. Thus, we can conclude he represents Mike. [14]

6. Lexicon. For example, when the text has USA and president we can have a lexicon to match this information - Donald Trump(he,singular). (From QA session 27/05.)

7. String feature. How many letters overlap of two mentions. For example, Donald Trump and Trump

5.3 Rounding up system design

5.3.1

Sieves

Pass 1-Speaker Identification. This process is about identifying speaker. For example, In direct quotations: Sam asked Mike, " Where did you buy your jacket? I really like it." I is the speaker, so it matches with Sam and Mike will refer with the addressee you. So first person pronouns will match the speaker. And second person pronouns will match addressee.([22],894)

Pass 2 and 3-Exact Match and Relaxed String Match. Pass 2: Only when the two mentions are exactly the same, they will be matched. For example *Apple CEO Tim Cook* and *Apple CEO Tim Cook*. Pass 3: After deleting the content following the head words, if these two mentions are the same, they are coreference. For example, *Tim Cook* and *Tim Cook, who is Apple CEO*.([22],894)

Pass 4 – Precise Constructs. There are some precise steps to match the two mentions. When two mentions are in same appositive construction (Apple CEO, Tim Cook said), when they have exact subject–object relation (Tim Cook is Apple CEO), when a noun is ahead of antecedent (CEO Tim), when is a relative pronoun(this is a street *which* has famous story), When they are acronym(University of Amsterdam-UvA) or demonym(Israel-Israeli), they are coreference. (Information form [22],895)

Pass 5, Strict head math is mainly about only when head words are exact same words(University of Amsterdam-University of Amsterdam), has compatible modifiers(nice Mike, good Mike), has same stop-words (*The Unites States, The States*), are not in i-within-i, don't have different locations and numbers(*China, South China*, *people, about 200 people*), the head words will match.

Pass 6, 7, 8, 9 are about head mach Or for relaxed head math. They basically give up the strict rules of Pass 5. there is one words match, the head words will match. For example, *Tim Cook, Tim, Apple CEO Tim Cook*, they all have *Tim*.

Pass 10-Pronominal Coreference Resolution. Number, gender, person, animacy, NER label, Pronoun distance(the distance of its antecedent is smaller than 3) are used in Pronominal Coreference Resolution.

5.3.2

We think Pass 1-3, and 5-9 are not suitable for these sentences. Pass 1 is about speaker and there is no speaker here. Pass 2-3 is about string match and 5-9 is about head match.

So we mainly focus on Pass 4 and 10.

C can match Pass 10-Pronoun distance. We can see the distance between Mark and he is shorter than 3 token.

F can match Pass 4-Predicate nominative. Because Subject to object. John to Mark- He to he.

I can match Pass 10-Pronoun distance. We can see the distance between They and chocolate is shorter than 3 token.

H and I may match Pass 10-gender. Because there are *it and they* in sentence.

Other sentence we don't see any match.

5.3.3

I think my features from 5.2.2 all have correspondence in Pass. For example: String feature matches Pass 2 and 3. Position feature matches Pass 10-Pronoun distance. Previous and following token feature match Pass 4 - Role appositive. Gender and number feature match 10-Gender and number.

Here I want to add a new feature- Capitalization. This feature was inspired by Pass 4- Acronym [22]. For example, Vrije Universiteit- VU. The first mention is maybe the whole word. But it may use abbreviation after that. Thus we need capitalization feature to detect the upper case and extract them. In this way, when VU occurs we know it is Vrije Universiteit.

6 Distributional Semantics

6.1 Measuring Similarity: Intrinsic Evaluation

6.1.1

6.1.2

We used predictive model-word2vec in the experiment because it is efficient and has high performance. The input of this model is text corpus and the output are word vectors. This vectors mean semantic similarity of words pairs. And these word vectors can be used as features in the machine learning model. For example, When we use the model to generate the most similar words to *cry*, the results of model show *('crying', 0.661), ('cries', 0.655)* . . . . The first is most similar words and the number is score of similarity. And it also can compare the word pairs' similarity. For example, the similarity of *'man', 'woman'* is *0.766*. While the similarity of *'man', 'dog'* is *0.309*.

In this part, we use *SimLex999*, which is human judgments of similarity scores of word pairs. And then we used our model to calculate the same word pairs' similarity scores. And Spearman Rho is used to compare the results of human and model. Theory of Spearman Rho is comparing of rank of same word pair in human and model scores. For example, *vanish-disappear* word pair ranks first in human score and ranks third place in model score. Thus the distance $d_i^2 = (3-1)^2 = 4$. Thus, the Spearman Rho is give by 2, where n is sample size.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \tag{2}$$

And the Spearman Rho is 0.4419 and p-value is $5.6 * 10^{-49}$, which means we need to reject the null hypothesis and two scores have high correlation.

6.1.3

In this part, we want to investigate the rank difference between the human score and model score. The rank difference is measured by $R_{human} - R_{model}$. So when Rank difference is positive it means model ranks higher than human ranks.

Table 6:

Pairs	Rank difference($R_{human} - R_{model}$)
acquire-get	-773
god-spirit	-606
motor-engine	-212
bathroom-kitchen	590
winter-summer	710
accept-reject	845

Table 6 shows part of result of rank difference. We can see human has high rank on *god-spirit,acquire-get,motor-engine* and model has high rank on *bathroom-kitchen,winter-summer,accept-reject*.

One observation is that antonym pairs of model rank are much higher than human rank, which means the model think antonym is similar to each other. For example, we can see *winter-summer*, the rank of model is 38 and human rank is 748. The possible explanation is that these antonym has similar function/roles in sentence or text. For example, *I like winter because of snow, and she like summer because of sun.* Since the model can not know the actual meaning of *winter and summer*. Model can only get the information from running text or bag-of-words[19]. But these two words are all the same except the meaning. Thus, the model lack this information and will just predict *winter-summer* are similar to each other. For human rank since human know exactly the meaning and *winter and summer* are opposite thing and can not be similar. Thus, if we want the model to be more accurate, we should import a feature about to distinguish the antonym, syntactic or dependency[19]. This could be a lexicon, and match word in sentences or text. Another explanation is maybe the text of input is too broad and it is hard for model to capture the similarity and but good for capturing relatedness. But for some pairs model has nearly the same ranks as human. For example, *top-side, attorney-lawyer,jar-bottle, sheep-lamb,. . . .*

6.2 Machine Learning: traditional features and embeddings

6.2.1 NERC

In this part, we start to explore how the individual features influence the machine learning system. The main task is to predict the named entity recognition. *'Token','Prevtoken','Cap','Pos','Chunklabel'* are used as features in the model. Token is spliting the whole sentence or text into single words. This is helpful for the system because it can detect whether the token has been previously observed as being part of a name is useful information to determine whether it is currently part of a name(from assignment feedback). And we found the result is *P: 76.6% R: 48.1% F1: 56.4%*. We can see the precision is quite high but Recall and F1 are relatively low.

Next feature is Prevtoken, which is previous token of current token and if there is no further token this feature is empty. This feature aim to elaborate previous token. For example, Dr. Mike and current token is Mike. We can still have a information of previous token Mike. And we get the result: *P: 62.8% R: 30.7% F1: 36.6%*. We see it shows worse result than only token feature.

Next feature is Cap means Capitalization, which contain lower letter, upper letter and so on. Cap is quite useful feature. For example, When the first letter is upper case this word is probably a name entity(China, Netherlands). And the result:*P: 9.1% R: 11.1% F1: 10.1%*. We can see only cap feature is not ideal.

Then, we use Pos-tag feature. Pos-tag can label the word with their own part-of-speech. In this way, the model can focus on noun, prop noun. And the result: *P: 12.8% R: 21.5% F1: 15.0%*. We can see it is better than only cap feature. But it still has bad performance.

The final feature is Chunklabel (phrases or constituents). This feature labels noun, verb and prepositional phrases(maximum two words). This could help to increase accuracy because name entity are usually not phrases. And the result: *P: 9.18% R: 11.1% F1: 10.1%*. We can see the result is extremely not ideal.

Thus next step is combine all these features.

6.2.2 Ablation Analysis

Furthermore, we did ablation analysis of features selection. Table 7 shows selection of label. We check every possible combination of these five features. Table 9 show the best combination of each number of features. And it is clear to see that when the features are Token, Prevtoke, there is high precision 86.6%. But the recall and f1 score are really low. And we can see the best combination of features are *Token, Prevtoken, Cap* or *Token, Prevtoken, Cap, Pos*. *Token, Prevtoken, Cap* have higher precision but lower recall and F1 score. While *Token, Prevtoken, Cap, Pos* have higher recall and F1 score. This indicates *Token, Prevtoken, Cap* are essential feature for NER. Because token help model to detect the whether the word is part of name entity and Capitalization is highly correlated with name entity(for example upper case). POS-tag help the model in a limited way.

Furthermore, it seems like Chunklabel has no influence on name entity, which indicates that the phrases are not that useful for name entity. Then we can remove it.

Table 7: Best feature selection

Number of feature	Best features	Best precision	Best recall	Best f1
1	Token	76.6%	48.1%	56.4%
2	Token, Prevtoken	86.6%	58.6%	68.5%
3	Token, Prevtoken, Cap	79.3%	72.1%	74.7%
4	Token, Prevtoken, Cap, Pos	77.8	73.5%	75.0%
5	Token, Prevtoken, Cap, Pos, Chunklabel	77.5%	73.5%	74.9%

6.2.3 Word Embeddings

Additionally, we use word embeddings to represent the token and to do the task of predicting the named entity recognition. One-hot representation has disadvantage that it can not describe the dependency and relation between words because it is a sparse matrix and can not store the information about dependency and relation. While word embeddings has this features. The word vectors generated by word embeddings contain semantic relations. The distance between the two word vectors measures the similarity of the words, and the added value of the two word vectors is also semantically added. [11]

The result shows when there is only Token in word embeddings the three scores are still not that great. While when adding Prevtoke in in word embeddings the result show greatly improvement and it is better than traditional features. And the best performance shows in mixed system. We can see that precision is nearly the same but recall and f1 score are improved greatly. We think this is mainly because the model incorporate the dependency and relation between words, which is provided by word embeddings and will help to finish NER because semantic feature is important for NLP.[11]

Table 8: Outcome of word embeddings

Best features	Best precision	Best recall	Best f1
Token	67.2%	63.4%	65.1%
Token, Prevtoken	77.1%	75.5%	76.2%
A mixed system	77.2%	78.6%	77.8%

6.2.4 Conclusion

Best feature for traditional feature is *Token, Prevtoken, Cap* or *Token, Prevtoken, Cap, Pos*. Word embeddings model show better performance than traditional model. And the best system is mixed traditional and word embeddings. Traditional model use one-hot representation, which is efficient but lacks the information of dependency and relation between words. Word embeddings has these semantic information, which will have better performance for NER task than traditional system. And combining two system is great idea.[11]

7 Module 3 Error propagation

For some sentences, I used ”we” as subject. But actually only me finish this assignment. The ”we” is just used for convenient!!!

7.1

For example:

1. Muscovy(NN) duck(NN) is very cute. 2. I successfully duck(VB) his attack

We can see the first duck is noun and it is name entity because it is one kind of duck. The second sentence is a verb. And it means avoid.

8 POS-Tagger performance

8.1 Analysis

We used result from MBT to generate POS-tag from NLTK. From result, there are total 203621 tokens and we see that there are 19465 difference of POS-tagging between MBT Tagger and NLTK. The error percentage is 9.56%. And we count the number of main difference between MBT Tagger and NLTK. We found 1836 differences of CD in MBT but JJ in NLTK('CD','JJ':1836). And 1253 difference of NN in MBT but NNP NLTK('NN','NNP':1253). And ('JJ', 'NNP'): 962, ('JJ', 'NN'): 673, ('NNP', 'JJ'): 731 and so on.

8.2 Interpretation

First we investigate what part-of-speech causes the main disagreement of NEr. We see ('JJ', 'NNP'), 806), (('NN', 'NNP'), 688), (('NNP', 'JJ'), 577), (('NNP', 'NN'), 399). We can see that 806 words in MBT are JJ but in NLTK are NNP. Then we further investigate how the POS-tag influence the named entity words. We generate the words which cause different prediction between MBT and NLTK. We find that most words are not have big influence on named entity recognition such as *that(85 different POS-tag)*. But some words do have influence on named entity recognition. For example, *Europen (27 different POS-tag)*. In NLTK, many *Europen* are MBT but in NLTK it is NNP. In this way the POS-tag will cause disagreement on NER because NNP is important features for NER.

8.3 Classification

In this part, we use the model from assignment 2. For MBT POS-tagger we have both tradition model and word embeddings. And then we have the different POS-tagger:NLTK data. We also use both tradition model and word embeddings to do the NER prediction. The features we used for both POS-tagger are *'Token','Prevtoken','Cap','Pos','Chunklabel'*.

Then we compare the result from NLTK and MBT as table 9 and 10. We see for both model the POS-tag from NLTK is slightly better than MBT and in general there is no big difference between two POS-tagger, which actually corresponds our conclusion from 6.2.2. We stated that the POS-tag features have little influence on this model for NER task. The possible explanation is that these two POS-tagger bring same certain error. For example, There are 962 part-of-speech in MBT:"JJ" but NLTK:"NNP". While there are 748 part-of-speech in MBT:"NNP" but NLTK:"JJ"(from my own code). We can see the difference is quite small and both part-of-speech prediction "error" are kind of similar, which leads to the result is similar. Another explanation is the POS-tag feature indeed have small contribution to the NER task, which means it will bring nearly nothing change on model or result.

Table 9: Best feature selection

Number of feature	Best features	Best precision	Best recall	Best f1
5-NLTK	Token, Prevtoken, Cap, Pos, Chunklabel	77.8%	73.7%	75.0%
5-MBT	Token, Prevtoken, Cap, Pos, Chunklabel	77.5%	73.5%	74.9%

Table 10: Outcome of word embeddings

Best features	Best precision	Best recall	Best f1
A mixed system-NLTK	77.4%	78.3%	77.8%
A mixed system-MBT	77.2%	78.6%	77.8%

9 Error Analysis Part 1

9.1 Analyze

First, we calculate how many wrong prediction of the model of 10% prediction of lstm. We find there are total 15603 token(delete ———-). And there are total 428 wrong prediction. The accuracy is 96%, which is quite high. Furthermore, We look at where is error mainly from. We find there are total 202 error from the prediction of location for both B-org and I-org, which takes 32.6% percentage of total error(202/618).

And we noticed that each sentence is divided by ————-. So we want to investigate whether length of sentence influence prediction. We calculate mean of the right prediction for different sentence length(token number). For example, *Amsterdam, Netherlands is a city.* There are total 7 tokens and golden label is *B-LOC O B-LOC O O O O*. If the prediction is *B-LOC O O O O O O*. The accuracy is (6/7). And we calculate average accuracy of each different sentence length. Figure 9.1 show our result. We can see around 40-60 tokens the accuracy drop quickly. **So our first hypothesis is the length of sentence can influence the accuracy of prediction.**

And then we look at the wrong prediction on specific named entity. We notice there are 222 out of 428(51.9%) wrong prediction of person's name(B-PER and I-PER).(result from my own code). Thus, **our second hypothesis is this model is bad at predict PER attributes.**

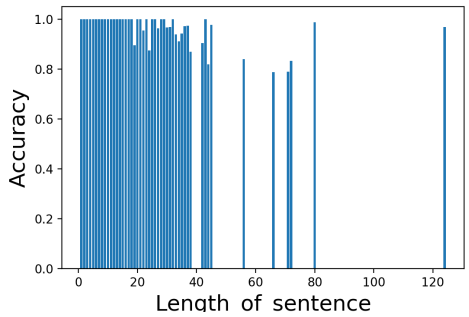

Figure 8: Accuracy of difference length

9.2 Operationalize the hypotheses

According to the paper[27], the hypothesis need to be precise and clear. Thus, we need to investigate the 10% dataset carefully.

For the first hypothesis, we carefully investigate the accuracy of different sentence length. We found out when sentences have more than 50 tokens the accuracy drops significantly. Thus, our first hypothesis is **model is bad at predicting the sentences which have more than 50 tokens.**

For second hypothesis, We separate the first name and last name, for example, we count Qinghe Gao are two names. And we found out as figure 9.2 shows when the name has more than 5 letter the accuracy is quite low. And this accuracy is calculate like: 1. Count the number of the length of the each name when gold label=*B-PER and I-PER*. 2 Count the number of wrong prediction of each name's length. 3. Accuracy is using (step 1 - step 2)/step 1. Thus, our second hypothesis is **model is bad at predicting the person name which have more than 5 letters.**(From my own code)

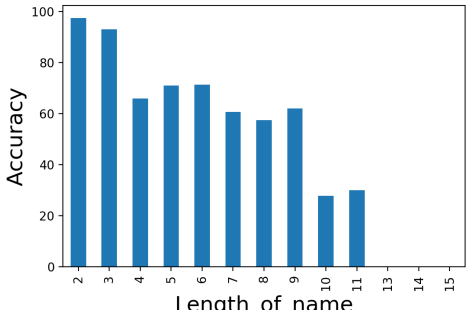

Figure 9: Accuracy of difference length of name. 10% dataset.

And We don't need to justify our hypothesis after checking the train and validation dataset manually. Because many sentences in train and validation dataset still have more than 50 tokens. Thus, the model will perform consistently on both experiment step and prediction step. And we also find that for second hypothesis there are many names which have more than 5 letters.

9.3 Test the hypotheses

Finally, we can verify our hypothesis on full LSTM test .

For the first hypothesis, we calculate the accuracy for each sentence which is divided by ————-. We use same step to calculate the accuracy. The accuracy is right prediction over total prediction. And then we average the accuracy of sentences which have same tokens. Figure 14(a) shows our result of first hypothesis. We can see when sentences have more than 55 tokens the accuracy drop quickly. Thus, we need to adjust our hypothesis. The model is bad at sentences which have more than 55 tokens.

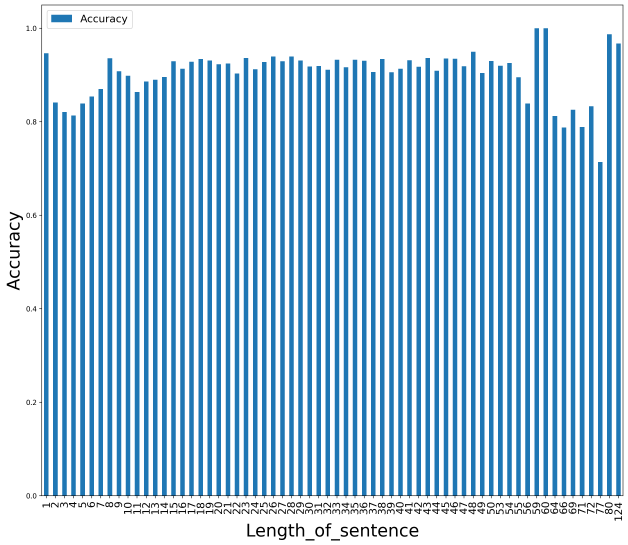

Figure 10: First hypothesis.

And for second hypothesis, first we count the whole wrong prediction. We found there are 4167 wrong prediction. Then we further count the number of wrong predictions on name(for both B-PER and I-PER), which is 1184. Thus, 28.4% wrong prediction on PER. Then, we further count the wrong prediction names which have more than 5 letters, which is 693. And 58.5% wrong prediction is the name which have more than 5 letters. And as figure 9.3, we can see that when length of name is bigger than 3 there is a clear drop on accuracy. And When length of name is bigger than 7 there is another clear drop on accuracy.

Thus, we could adjust that the model is bad at predicting the name which have more than 7 letters.(Data from my own code)

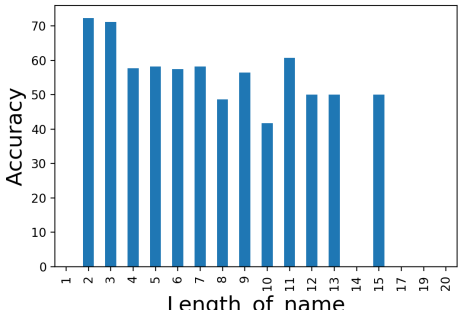

Figure 11: Accuracy of difference length of name. Total dataset.

10 Manipulating the input

In this part, we do adversarial examples in this part. And still we want to test hypothesis: the model is bad at predicting the person name which have more than 7 letters. And as for adversarial examples, we need to manipulate the **input without changing golden labels**. Thus, our steps show as follow:

- Generate name list. In this step, we need to generate a name list which contains the names that have less than 7 letters(include 5 letters). This name list could extract the name from train, validation or just test dataset.

- Replace name. In this step, we randomly pick up a name in name list and substitute it with 7-more-letter name in input dataset. We also can do another way around. Like substituting all the 7-less-letter name with the name which have more than 7 letters.
- Then analyze the prediction

The reason we did this is our hypothesis is the model is bad at predicting the name which have more than 7 letters. So if we let all the name have less than 7 letter and calculate the accuracy, it will verify our hypothesis.

The adversarial examples show as data-adversarial dataset. Here are some example in the dataset. The adversarial examples show as data-adversarial dataset. Here are some example in the dataset.

- Original. when Uzbek striker Igor **Shkvyrin** took advantage of a misdirected . . .
- Adversarial. when Uzbek striker Igor **Dion** took advantage of a misdirected . . .

The whole data set is in zip file.

11 Interpretation of classification results (new data and model)

11.1 Data

1. We found there are 33557 sentences in training data, 7194 sentences in test data and 7193 sentences in validation data.

2. Figure 11.1 shows the named entity labels and number of labels for training data. This table is from my own code.

Label	O	B-geo	B-tim	B-org	I-per	B-per	I-org	B-gpe	I-geo	I-tim	B-art	B-eve	I-art	I-eve	B-nat	I-gpe	I-nat
counts	621876	26343	14219	13950	12220	11980	11540	11103	5236	4632	299	240	232	185	137	137	35

Figure 12: Information of label of train data.

3. And we see 734351 vocabularies in training data, 156324 vocabularies in test data and 157968 in validation data. And 35180 vocabularies in word.txt.

11.2 Model

11.2.1

The accuracy is the correct prediction percentage, which are the entries on diagonal divided by total number. In this case: For frozen embeddings, accuracy: $147776 * 100/156234 = 94.58\%$. For finetuned embeddings,accuracy: $149763 * 100/156234 = 95.86\%$.

The we calculate the precision, recall and f1 score. The way to calculate these three method. And we only use two categories as example(table 12):

Precision = True Positives / (True Positives + False Positives)

Recall = True Positives / (True Positives + False Negatives)

F1 Score = 2 * (Precision*Recall)/(Precision+Recall)

Table 11:

	0	1
0	True Positives	False Positives
1	False Negatives	True Negatives

And here is the result of two model. We can see that finetuned embeddings has better result for all three score. And all the results are calculate by code(my own code and code from assignment 2).

Table 12:

Model	Precision	Recall	F1 score
Frozen	63.4%	48.6%	52.7%
Finetuned	69.2	57.0%	60.0%

11.2.2

Then we calculate the majority baseline that labels everything as ”O”.

For the both, we calculate the accuracy is 84.6%. And precision is 4.98%, recall is 5.88% and F1 score is 5.39%.

We can see that the accuracy of two methods are little bit higher than baseline. But when it comes to precision, recall and F1 score. The baseline show extremely bad performance. We can see all the three score are very small, which make senses because we label everything as ’O’ which loses predictions on each label.

When compared to the baseline both models show great improvement on accuracy, precision, recall and F1 score.

11.2.3

As we can see, the accuracy, precision, recall and F1 score from 11.2.1 show that finetuned model is better than frozen model. And also because finetuned model knows how to learns to finetune the embeddings. Thus, we choose finetuned model.

And from figure 13 we can see the biggest difference of corrected prediction is *B-geo*. Frozen model predicts 412 fewer correct *B-geo* than finetuned model. And also for *B-per* frozen model predicts 334 fewer correct *B-per* than finetuned model. That’s indicates that the frozen model is not good at geographical Entity and person name.

Gold	B-art	B-eve	B-geo	B-gpe	B-nat	B-org	B-per	B-tim	I-art	I-eve	I-geo	I-gpe	I-nat	I-org	I-per	I-tim	O
B-art	0	0	13	0	0	10	1	1	0	0	0	0	0	2	1	0	38
B-eve	0	10	2	3	0	5	2	0	0	0	0	0	0	1	0	1	13
B-geo	0	0	4539	46	1	266	58	30	0	0	67	0	0	18	25	6	511
B-gpe	0	0	65	2307	0	20	5	3	0	0	4	0	0	4	0	0	84
B-nat	0	0	6	0	3	1	1	1	0	0	0	0	4	0	0	1	16
B-org	0	0	543	38	0	1599	102	7	0	1	12	0	0	47	50	5	654
B-per	0	0	70	10	0	104	1539	3	0	0	5	0	0	38	109	4	596
B-tim	0	0	72	0	0	22	2	2418	0	0	1	0	0	3	1	40	546
I-art	0	0	0	0	0	0	0	1	0	0	3	0	0	12	2	0	17
I-eve	0	1	0	0	0	2	0	3	0	4	1	0	0	11	0	0	11
I-geo	0	0	39	4	0	6	2	1	0	0	790	0	0	56	24	0	100
I-gpe	0	0	0	12	0	1	0	0	0	0	4	14	0	2	0	0	5
I-nat	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0	6
I-org	0	0	46	12	0	72	28	0	0	0	123	0	0	1601	141	3	553
I-per	0	0	6	1	0	14	119	0	0	0	28	0	0	109	1962	1	281
I-tim	0	0	14	0	0	3	1	81	0	0	0	0	0	8	2	586	292
O	0	0	297	38	5	385	238	231	0	1	78	0	5	220	158	113	130404

(a) Confuse matrix of frozen model

Gold	B-art	B-eve	B-geo	B-gpe	B-nat	B-org	B-per	B-tim	I-art	I-eve	I-geo	I-gpe	I-nat	I-org	I-per	I-tim	O
B-art	4	0	18	1	0	11	4	0	0	0	0	0	0	3	0	0	25
B-eve	0	10	3	2	0	6	1	0	0	0	0	0	0	1	0	1	13
B-geo	2	0	4951	41	1	215	62	10	0	0	65	0	0	18	17	1	184
B-gpe	0	0	80	2379	0	10	3	0	0	0	4	0	0	3	1	1	11
B-nat	0	0	3	0	12	5	1	0	0	0	0	0	2	0	0	1	9
B-org	1	2	616	36	0	1785	122	12	0	3	11	0	0	36	40	1	393
B-per	0	0	93	5	1	117	1873	7	0	0	6	0	0	22	97	3	254
B-tim	0	1	67	0	0	19	6	2534	0	0	1	0	0	0	0	30	447
I-art	0	0	0	0	0	0	1	1	1	1	10	0	0	11	1	1	8
I-eve	0	1	0	0	0	0	1	2	0	6	2	0	0	12	0	1	8
I-geo	0	0	37	3	0	6	0	0	0	0	859	0	0	57	24	0	36
I-gpe	0	0	0	7	0	1	0	0	0	0	4	25	0	1	0	0	0
I-nat	0	0	0	0	1	0	0	0	1	0	2	0	1	1	0	1	3
I-org	0	0	50	6	0	59	39	0	0	4	168	0	0	1795	133	2	323
I-per	0	0	9	0	0	24	110	0	0	0	58	0	0	91	2138	5	86
I-tim	0	2	16	0	0	2	2	88	0	1	3	0	0	4	0	643	226
O	1	2	256	19	6	306	227	214	0	8	48	0	4	169	49	117	130747

(b) Confuse matrix of finetuned model

Figure 13: result

11.2.4

We use finetuned model to perform the analysis. First we investigate the accuracy of each label. We can see from figure 14 finetuned model is good at predicting *Geopolitical Entity*. We can see the accuracy of *B-gpe* is 95.47% and precision, recall and f1-score are also really high-95%. And also we can see the model is bad at predicting Artifact. The accuracy of *B-art* is extremely low, which is 6.06%.

Thus, we can use this model to predict geography class or some text mining about geography show: National Geographic. The training data should delete the artificial name entity because this model is bad at artificial name entity.

Label	counts	correct	Accuracy/%
B-art	66	4	6.06
B-eve	37	10	27.03
B-geo	5567	4951	88.93
B-gpe	2492	2379	95.47
B-nat	33	12	36.36
B-org	3058	1785	58.37
B-per	2478	1873	75.59
B-tim	3105	2534	81.61
I-art	35	1	2.86
I-eve	33	6	18.18
I-geo	1022	859	84.05
I-gpe	38	25	65.79
I-nat	10	1	10.00
I-org	2579	1795	69.60
I-per	2521	2138	84.81
I-tim	987	643	65.15
O	132173	130747	98.92

(a) Accuracy of finetuned model of each label

	precision	recall	f1-score	support
B-art	0.50	0.06	0.11	66
B-eve	0.56	0.27	0.36	37
B-geo	0.80	0.89	0.84	5567
B-gpe	0.95	0.95	0.95	2492
B-nat	0.57	0.36	0.44	33
B-org	0.70	0.58	0.63	3058
B-per	0.76	0.76	0.76	2478
B-tim	0.88	0.82	0.85	3105
I-art	0.50	0.03	0.05	35
I-eve	0.26	0.18	0.21	33
I-geo	0.69	0.84	0.76	1022
I-gpe	1.00	0.66	0.79	38
I-nat	0.14	0.10	0.12	10
I-org	0.81	0.70	0.75	2579
I-per	0.86	0.85	0.85	2521
I-tim	0.80	0.65	0.72	987
O	0.98	0.99	0.99	132173
accuracy			0.96	156234
macro avg	0.69	0.57	0.60	156234
weighted avg	0.96	0.96	0.96	156234

(b) Score of finetuned model of each label

Figure 14: finetuned result. The left one is from my own code and the right one is from a function classification-report.[10]

11.3 Error Analysis Part 2

11.3.1 Data Analysis

Our first hypothesis is **model is bad at predicting the sentences which have more than 50 tokens.** Second hypothesis is **model is bad at predicting the name which have more than 7 letters.**(From my own code)

Then there is a new data, so we want to investigate whether these two hypothesis can also apply into this result.

First, we used same method as 9.1 to check different length of sentence's accuracy. As figure 16 shows, we can see the accuracy is consistent among different sentence length. Thus, our hypothesis need to be abandoned.

Next we check our second hypothesis. We also used the same method as 9.2. As figure 15 shows the different observation from hypothesis. When length of name is smaller than 4 the accuracy is even zero. But we think this is may because the sample size. And we will verify it later. Thus, we need to adjust our hypothesis: **model is bad at predicting the person name which have less than 4 letters.**

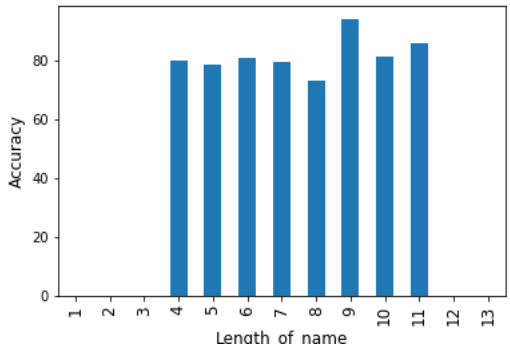

Figure 15: New data different length of name.10% dataset

Thus, we need to come up with a new hypothesis. We find out the main errors are caused by prediction on organization. There are total 202 out of 618(this is total wrong prediction) wrong prediction. So we formulate a new hypothesis this model is bad at predicting organization name(B-org,I-org). All the data analysis is from my own code.

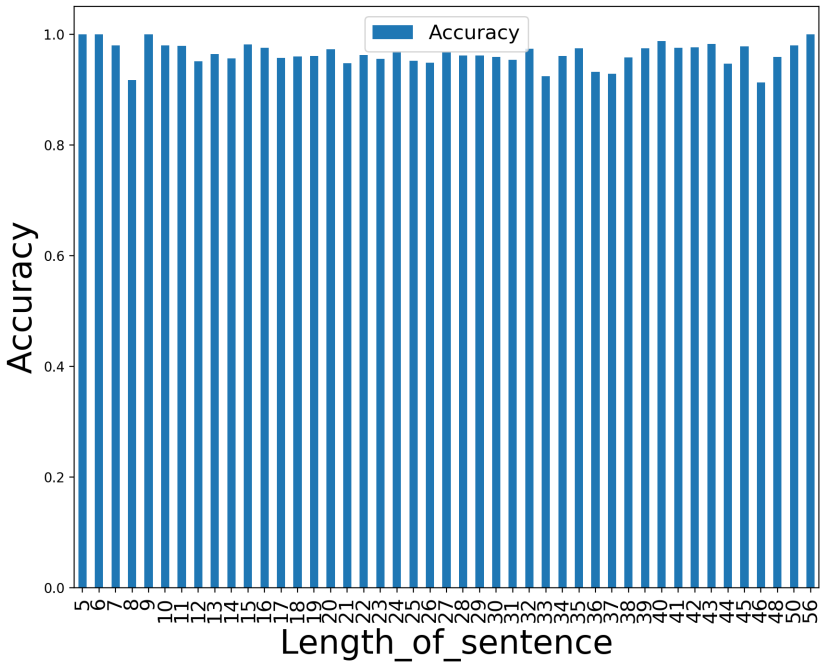

Figure 16: New data different sentence length

11.3.2 Hypotheses

The first hypothesis in this part is **model is bad at predicting the Person name which have less than 4 letters.** And we check out the training and validation dataset manually and find out there are many names which have less than 4 letters.

Then we need to elaborate the second hypothesis. We also investigate whether the number of letters of organization influence the accuracy. And there are total 202(32.68%) out of 618 wrong predictions on organization. This is a large error. And we used the same step as the first hypothesis. 1. Count the number of the length of the each origination name when goldlabel=B-org and I-org. 2 Count the number of wrong prediction of each origination name's length. 3. Accuracy is using(step 1 - step 2)/step 1. And we can see when length of organization name is smaller than 6. The accuracy drops quickly. And we can also see that when length of organization name is bigger than 9 the accuracy is nearly zero. But we think this is may because the sample size. And we will check it later. So our second hypothesis is **model is bad at predicting the organization name which have less than 6 letters.** All the data analysis is from my own code.

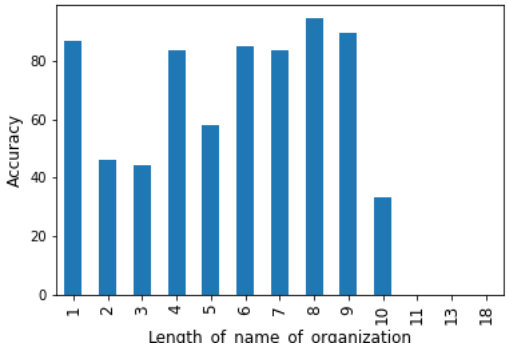

Figure 17: New data different length of organization name

11.3.3 Hypotheses Testing

Furthermore, we use whole dataset to verify our hypothesises. As figure 18 shows our first hypothesis does not apply to the whole dataset. The accuracy is low at length of person name is 1. And when the length of person

name approaches to more than 13 the system seems crush and cannot have correct prediction. It means the 10% dataset brings some biases to our hypothesis. For example, all the wrong prediction of person name which has less than 4 letter accidentally occur in 10% dataset. Thus, we think **this system is not good at prediction the name which has more than 13 letter.**

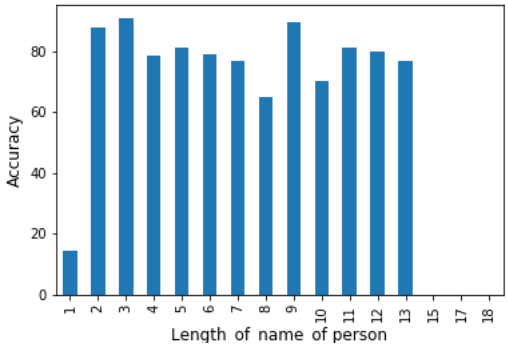

Figure 18: New data different length of person name. Whole dataset

Next, we check out the second hypothesis. As figure 19 shows, we can see the accuracy is comparatively lower when the length of name of organization is smaller than 6. Furthermore, we see when the length of name of organization is bigger than 9, the accuracy is quite high. This means the reason that the accuracy is almost zero when the length of name of organization is bigger than 9 is sample size. **Model is bad at predicting the organization name which have less than 6 letters.**

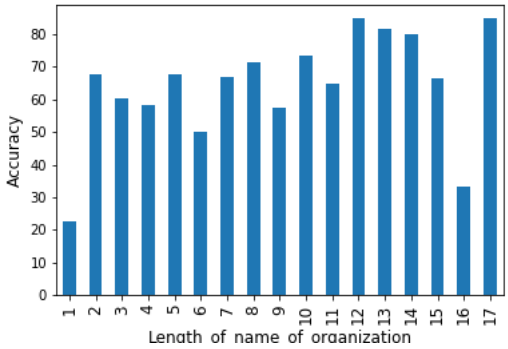

Figure 19: New data different length of organization name. Whole dataset

11.3.4 Comparison

In this part, we want to compare the main error of two models. The figure 20 shows the main type of error of two models. First we illustrate meaning:(('O', 'B-ORG'), 396). 'O' means gold label, 'B-ORG' means prediction label and 396 means error occurrence. We can see the common thing is that both model have large errors on organization name. And also for random embeddings we can see it is also bad for person name. While pretrained embeddings is bad at predicting time and geography. So we can conclude that the main challenge of NER is predicting organization name, which makes sense because we all know there are a sea of organization names in the word and everyday even comes up a sea of new organization names. And it is hard and changeling for model to catch and detech all the name.

And to further compare the model we need to work on same data set on different model or different dataset on same model. Now everything is different and it is hard to interpret deeply.

All the picture is from my own code.

```
[(('O', 'B-ORG'), 396),
 (('B-PER', 'O'), 388),
 (('B-ORG', 'O'), 344),
 (('I-PER', 'O'), 330),
 (('O', 'B-MISC'), 211)]
```

(a) Main error of CoNLL data with random embeddings

```
[(('B-org', 'B-geo'), 616),
 (('B-tim', 'O'), 447),
 (('B-org', 'O'), 393),
 (('I-org', 'O'), 323),
 (('O', 'B-org'), 306)]
```

(b) Main error of Kaggle data with pretrained embeddings

Figure 20: Result

References

[1] Assignment 2. https://github.com/cltl/ma-hlt-labs/).

[2] casemaker. https://courses.umass.edu/phil595s-gmh/book/04%20-%20Case-Marking%20-%20Syntax.pdf.

[3] Chinese. http://esl.fis.edu/grammar/langdiff/chinese.htm.

[4] Constituent. https://en.wikipedia.org/wiki/Constituent_(linguistics).

[5] dependency. http://nlpprogress.com/english/dependency_parsing.html.

[6] Evidentiality. https://en.wikipedia.org/wiki/Evidentiality.

[7] Hebrewlanguage. https://www.engheb.com/htm/blog-hebrew_morphology_made_simple.htm.

[8] Lemma. https://www.quora.com/What-is-lemmatization-in-NLP.

[9] oxfordbibliographies. https://www-oxfordbibliographies-com.vu-nl.idm.oclc.org/view/document/obo-9780199772810/obo-9780199772810-0118.xml.

[10] scikit-learn. https://github.com/scikit-learn/scikit-learn/blob/fd237278e/sklearn/metrics/_classification.py#L1825.

[11] Word embedding. https://blog.csdn.net/bqw18744018044/article/details/83722890.

[12] Inflectional and derivational morphemes. https://semanticsmorphology.weebly.com/inflectional-and-derivational-morphemes.html, 12 2017. Accessed: 1-12-2019.

[13] BENDER, E. M. Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. *Synthesis lectures on human language technologies 6*, 3 (2013), 1–184.

[14] BENGTSON, E., AND ROTH, D. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (2008), pp. 294–303.

[15] BOHNET, B., BJÖRKELUND, A., KUHN, J., SEEKER, W., AND ZARRIESS, S. Generating non-projective word order in statistical linearization. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2012), pp. 928–939.

[16] CHAHUNEAU, V., SCHLINGER, E., SMITH, N. A., AND DYER, C. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), pp. 1677–1687.

[17] COHEN, Y., BEN-SIMON, A., AND HOVAV, M. The effect of specific language features on the complexity of systems for automated essay scoring.

[18] EISENSTEIN, J. Natural language processing, 2018.

[19] HILL, F., REICHART, R., AND KORHONEN, A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics 41*, 4 (2015), 665–695.

[20] KOLCHYNA, O., SOUZA, T. T., TRELEAVEN, P., AND ASTE, T. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv preprint arXiv:1507.00955* (2015).

[21] KU, L.-W., HUANG, T.-H., AND CHEN, H.-H. Using morphological and syntactic structures for chinese opinion analysis. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3* (2009), Association for Computational Linguistics, pp. 1260–1269.

[22] LEE, H., CHANG, A., PEIRSMAN, Y., CHAMBERS, N., SURDEANU, M., AND JURAFSKY, D. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics 39*, 4 (2013), 885–916.

[23] LI, J., SUN, A., HAN, J., AND LI, C. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[24] NG, H. T., AND LOW, J. K. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (2004), pp. 277–284.

[25] PAROUBEK, P. Evaluating part-of-speech tagging and parsing patrick paroubek. In *Evaluation of Text and Speech Systems*. Springer, 2007, pp. 99–124.

[26] PARSING, C. Speech and language processing.

[27] WU, T., RIBEIRO, M. T., HEER, J., AND WELD, D. S. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), pp. 747–763.