Coreference Resolution of concepts in clinical documents

Bharath Gunari, Hrishikesh Bajad, Raj Sadaye, Rushikesh Sargar, Shubham Gondane

Abstract

Coreference information has been shown to be beneficial in a number of NLP applications including Information Extraction, Text Summarization, Question Answering and Machine Translation. Coreference resolution of concepts, although a very active area in the natural language processing community, has not yet been widely applied to clinical documents. Clinical text is unique because it requires significant use of domain knowledge to support coreference resolution. The system is developed to find concept from natural language in Electronic Health records(EHR) and resolve coreference between those concepts.

1 Introduction

Coreference resolution is one of the important task of Natural Language Processing (NLP) identifying which noun phrase (or mentions) refers to which real world entity or concept in the text. Anaphore is a NLP term used to represent the relation between the linguistic expression where the interpretation of one linguistic expression (real world entity/ the anaphor) is dependent on the interpretation of another (the antecedent). When the anaphor and the antecedent point to the same referent in the real world. They are termed coreferential. Coreference resolution is important for information extraction. So in this project we aim to perform coreference resolution on Electronic Health Records. We try to form chains of entity mentions which point to same real world entity.

The exponential increase in the volume of Electronic Health Records(EHR) has created a huge opportunity for clinical research and practice. Coreferent relations are common in natural language discourse in both EHRs and in general domains. For example " There"s a cyst in her kidney, which is causing her a lot of pain", the pronoun "which" refers to the phrase "the cyst in kidney" . This example shows how the pronoun is a coreferent to the noun phrase. In other cases noun phrases can be coreferent as well, like "kidney stone" and "Nephrolithiasis". The real world object mentioned in a discourse by noun phrases or pronouns called entities and the phrases or pronouns appearing in discourses called mentions. Mentions can be a word or a phrase.

The application to Natural Language Processing(NLP) to this unstructured EHRs directly compliments the structured EHR data and hence does not produce any valuable information. The application of NLP is hampered due to the poor performance of relation detection tasks. For example lacking the coreference resolution may lead to misclassification in named entity recognition, patient risk prediction, cohort identification, clinical decision support and other clinical applications. Coreference serves the critical role of linking related information together. Attributes, temporal descriptions and contextual information necessary for understanding the conditions, symptoms and treatments occurred are spread across the discharge summary and we require coreference resolution for accurate interpretation of the data. Armed with a textual coreference resolution system, a higher-level system can resolve coreference between the narrative notes and the structured data to yield a much richer picture. This system can link a detailed prescription and laboratory data from EHR with the textual mentions in a clinical note.

The most influential learning based model for classifying is a mention pair model, which is operated by training a model for classifying whether two mentions are coreferent or not. But men-

tion pair have two problems that hinder the performance of such a system. First, it only informs us about its relation with its active mention but not how well the antecedent is relatively better or worse than others. Second, the information extracted from the two mentions is not enough for making an informed coreference decision. Hence this leads to shallow features rather than features having semantic information.

2 Related Work

Coreference resolution has been studied for years in NLP. Approaches have included both supervised and unsupervised methods. A common approach for coreference resolution is determining the best antecedent for every mention. Ng and Cardie studied many methods to choose best previous mentions, some methods included Closest Link and Best-Link which performed better than other methods. Zheng provided a good literature review for coreference resolution in clinical data, and they use a support vector machine based approach trained on syntactic, semantic and surface features. Wang evaluated coreference for the words "it", "this", and "that" within 1000 sentences taken from a clinical text, they used a rule based approach to get results in the range of 90% to 94%.

Rink implemented a multiple pass sieve approach similar to Raghunathan , this method involves multiple independent models for each resolving coreference which are executed in succession. Each model makes a coreference decision on pairs of concepts from the natural language text. Rather than considering all possible pairs of concepts from the text, each model has its own selection criteria for choosing a subset of those pairs on which to make the decisions. When there model was evaluated by CEAF they got 0.89 as F-measure for i2b2 dataset.

Davy studied the about the factors that contribute to coreference resolution in bayesian nets. They implemented a two step approach, first anaphoricity resolution followed by coreference resolution. Also they studied the effects of Noisy features, machine learning models, feature selection, anaphoricity accuracy and search window. They came to a conclusion that noisy features are a critical factor in the performance of the system, bayesian nets were able to represent the dependencies between the different mentions, feature selection did not add much improvement, anaphoricity accuracy hugely impacted the performance of the system and the search window size was hugely dependent on the contextual information.

Based on previous work on coreference resolution, supervised models perform well given a sizeable training corpus. Also previous studies suggest considering dependencies between pairs on a global level could highly increase the performance of the system.

3 Overview of task and data characteristics

The task of coreference resolution in clinical documents can be briefly divided into the following subtasks:

1. Finding concepts from natural language text.
2. Finding coreferent pairs for the discovered concepts
3. Forming chains using the coreferent pairs.

We have considered 5 major types of concepts:

- PERSON
- PRONOUN
- PROBLEM
- TEST
- TREATMENT

Every concept is identified by its phrase string , the start and end position of the concept and the concept type. For example:

c="patient" 42:5 42:5||t="person"

Here, The phrase string is patient. This concepts starts on line 42 of the document and has only one word in it hence the starting and ending word numbers are the same. Also the type for this concept is "person"

Examples:

According to the data present in the i2b2 dataset the person concept type includes proper noun, common noun or a personal pronoun(he, she, his, her etc.). The proper noun or common noun refer to a person.

For example:

c="dr. mielke" 101:5 101:6||t="person"
c="she" 99:0 99:0||t="person"
c="patient" 79:37 79:37||t="person"

A pronoun concept type includes any non-personal pronoun present in the document. For example:

c="they" 80:6 80:6||t="pronoun"
c="it" 95:5 95:5||t="pronoun"
c="they" 84:10 84:10||t="pronoun"

Problem concept type includes diseases, suffering, pain, grievances, etc that are present in the text. For example:

c="hallucinations" 94:9 94:9||t="problem"
c="dilated left intrarenal collecting system and proximal ureter" 83:32 83:39||t="problem"
c="asthma" 30:13 30:13||t="problem"

Test concept type includes any diagnosis task that is mentioned in the text. For example:

c="a bronchoscopy" 84:15 84:16||t="test"
c="her cardiac examination" 91:0 91:2||t="test"
c="an echocardiogram" 67:0 67:1||t="test"

The treatment concept type includes any procedure or medication that is mentioned in the text. For example:

c="cisplatin chemotherapy" 99:25 99:26||t="treatment"
c="her steroids" 71:0 71:1||t="treatment"
c="proventil inhaler" 103:4 103:5||t="treatment"

4 Initial Approach

4.1 Overview of Approach

Initially the approach we intended to follow, was to use 2 neural networks. One would be used to classify the phrases from natural language text into concept types The second would be used to identify coreferent pairs based on the features identifies from the concepts. The coreference chains would be identified using relations between coreferent pairs.

For extracting phrases from text, we used a library called SENNA. SENNA is a software which outputs a host of Natural Language Processing (NLP) predictions: part-of-speech (POS) tags, chunking (CHK), name entity recognition (NER), semantic role labeling (SRL) and syntactic parsing (PSG).

Noun phrase chunking was used for finding phrases which could potentially be concepts. Since the phrases are to be fed to a neural network, there is a need to represent the phrases in vector form. To represent these phrases in form of vectors we used Global Vectors for Word representation. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. GloVe presents us with two options :Use the pre-trained vectors or train the GloVe model with a custom corpus. We used the documents from i2b2 training data and 1500 documents from the MIMIC-III dataset to train the Glove model.

4.2 Experimental Details of initial approach

Initially we had generated GLoVe representation for all the phrases from the i2b2 train dataset namely Beth Train and Partners Train. Beth train has 115 documents and Partners train has 136 documents. The dimension of GLoVe was kept 20. The neural network which we used was as follows. Size of the dataset was 35059 x 20, where 35059 is the number of phrases found in the docs file of the i2b2 dataset.

4.2.1 Approach 1

The input to the neural network was a 18791 x 20 dimensional matrix. Next 10 fully connected dense layers were used with each layer consisting of 500 neurons. The output layer was a softmax with 5 neurons each for 5 classes or concept types.

The entire dataset was randomly split into training and testing sets, with the split being 33 percent of training set. The training set was then randomly split into a validation set, with the split being 20 percent of training set

Size of Training set: 18791 approximately
Size of Validation set: 4697 approximately
Size of Testing set:11569 approximately

We used Stochastic gradient descent (SGD) optimizer with the following hyperparameters:

Learning rate = 0.01
Decay=1e-6
Momentum=0.9
Nesterov=True

The model was trained using a batch size = 32 and number of epochs = 50.

This model was trained by varying parameters like number of layers, number of neurons per layer, batch sizes and number of epochs. Additionally we tried using different values of learning rate and Decay rate. But the highest precision we got was 27 percent approximately, with recall being around 26 percent.

Owing to this drawback we decided to increase the dimensions of the GLoVe representation to 50. Whereas the rest of the network parameters remained same

4.2.2 Approach 2:

We even tried using a different optimizer like Adam optimizer with the following hyperparameters -

Learning rate = 0.01
Beta1 = 0.9
Beta2 = 0.999
Decay = 0.0

The precision and recall values after using Adam Optimizer decreased to around 19 percent. Therefore we decided to go with SGD as it resulted in better performance.

4.2.3 Approach 3:

In order to get better performance out of the network we decided to add data from the MIMIC-3 dataset. We select 400 documents from the MIMIC-3 dataset. Concepts were extracted using the CLAMP software. We generated the GLoVe representations for these extracted concepts with dimension as 50. Now the dataset was of the dimension 147632 x 50.

The entire dataset was randomly split into training and testing sets, with the split being 33 percent of training set. The training set was then randomly split into a validation set, with the split being 20 percent of training set

Size of Training set: 98913 approximately
Size of Validation set: 19782 approximately
Size of Testing set:48718 approximately

The input to the neural network was a 98913 x 50 matrix Next 12 fully connected dense layers were used with each layer consisting of 1000 neurons. The output layer was a softmax with 5 neurons each for 5 classes or concept types.

We used Stochastic gradient descent optimizer with the following hyperparameters:

Learning rate = 0.01
Decay=1e-6
Momentum=0.9
Nesterov=True

We tried changing the batch size and number of epochs. We got the following results:

Batch Size	Epochs	Precision	Recall	F1	Accuracy
32	50	0.429	0.394	0.410	0.4192
50	50	0.43	0.403	0.416	0.422
64	50	0.433	0.4034	0.417	0.423
50	75	0.427	0.406	0.416	0.420
128	75	0.452	0.435	0.443	0.44
256	75	0.442	0.419	0.430	0.43

Figure1: table containing experimental results for neural network based approach

The final model which gave us best result for the evaluation metrics was the one with a batch size = 128 and number of epochs = 75. The precision and recall which we got was around 45.21 percent and 43.59 percent approximately respectively.

4.3 Final approach

Since the results from the neural network based method were not promising we decided to change the method for discovering concepts from Natural Language text. We implemented a method that would aggregate various tools to extract concepts from the text,then form mention-pairs from the concepts. Later doing feature engineering on the concepts we will apply a classifier that checks whether a mention-pair is valid or not.

To extract concepts related to the medical domain, i.e., problem, test and treatment, we used the CLAMP tool. The CLAMP System is a comprehensive clinical Natural Language Processing software that enables recognition and automatic encoding of clinical information in narrative patient reports. In addition to running a clinical concept extraction pipeline as well as an annotation pipeline, the individual components of the system can also be used as independent modules. The system lends itself for diverse applications in a broad range of clinical domains.

CLAMP contains a module called the Named-Entity Recognition(NER) pipeline which extracts phrases that are of types problem, test and treatment. The NER pipeline takes as parameters an input folder containing natural language text documents, a folder to store the output files. A sample output after running clamp on one of the documents :

NamedEntity	1943	1952	semantic=problem	assertion=absent	ne=tamponade
NamedEntity	1992	2007	semantic=problem	assertion=present	ne=cervical cancer
NamedEntity	2062	2071	semantic=problem	assertion=present	ne=malignant
NamedEntity	2144	2156	semantic=treatment	assertion=present	ne=further care
NamedEntity	2295	2314	semantic=treatment	assertion=present	ne=right ureteral tube

For finding non-personal pronouns in the text we used POS tagging from the python Natural Language Toolkit(NLTK). For finding concept type person we used Stanford NER tagger. We used a python wrapper of the Stanford NER tagger. The tagger classifies phrases into the following types: Location, Person, Organization, Money, Percent, Date, Time. We extracted the phrases which were classified as Person.

For every concept phrase discovered, a line number with start and end of the phrase was appended. The output of the concept generation phase was in the following format:

any anti retroviral therapy|22:4 22:7|treatment

In the pipe separated line the first string represents the concept string(phrase). The concept starts at line 22 word number 4 and ends at line 22 word number 7. The concept type is treatment.

We were also successful in extracting temporal concepts from text. We used Stanford Temporal Tagger for doing this. SUTime library was used for this. SUTime is a library for recognizing and normalizing time expressions. That is, it will convert next wednesday at 3pm to something like 2016-02-17T15:00 (depending on the assumed current reference time). SUTime is available as part of the Stanford CoreNLP pipeline and can be used to annotate documents with temporal information. It is a deterministic rule-based system designed for extensibility.

Output of the temporal concept detection module:

eight hours,temporal
eight hours,temporal
24 hours,temporal
years,temporal
1990,temporal
past,temporal
now,temporal
the future,temporal

But there was no training data related to temporal concepts in the i2b2 datasets. Hence we were unable to find coreferent time mentions in the data.

The phrases extracted need to be annotated with line numbers and their position or index in the line. For example: Each concept is annotated in the following way. The elements are separated by "|"

a rehabilitation hospitalization|18:4 18:6|treatment

The first element is the actual concept
The second element is the line number and the position
Where, 18 is the line number
4 is start index of the concept in the line
6 is the end index of the concept in the line
The third and the last element is the concept type.

To do this we had to look at the EHR documents provided in the i2b2 dataset and search for the phrases in these files. For each phrase extracted from a particular EHR document ,we then extracted the line number in which a particular phrase was found and it"s position in the line. All the lines where a particular phrase was found were considered for annotation.

4.3.1 Generating Mentions

As we extracted different concepts in the previous steps, in order to identify the chains of coreferential words from the documents we will try to find the pairs of words which are coreferential with

each other and such pairs would be used at the end to create chains out of them.

Approaches to find coreferential pairs : There could be multiple ways to find out the coreferential pairs from the document,

- Generate pairs one by one based on the similarities of words based on the context they are used in
- Generate all possible pairs from the document and remove the ones which are not coreferential , this approach ensures that we are considering all the possible cases and would help to achieve higher values of recall.

In our model we have used uses the 2nd approach mentioned above, generate all possible pairs with some logical constraints on it and reject the pairs using classifier to get the co-referent pairs.

Approach of making all possible mention pairs from all the concepts would result in having lot of pairs and which might consist a high number of pairs which are not co-referent. In practice if we have n concepts of each concept type, creating all combination pairs would result in n6 pairs, and many of them would be not related with each other.

For example:
end stage liver disease|12:10 12:13|problem
hepatitis C cirrhosis|12:16 12:18|problem
fluid balance|40:2 40:4|test
The patient "s INR|41:0 41:3|test
red cells|42:15 42:16|treatment
platelets|42:13 42:13|treatment
patient|12:1 12:1|person
his|38:24 38:24|pronoun
it|39:24 39:24|pronoun
Lynda|14:1 14:1|person
doctor|17:1 17:1|person
She|19:1 19:1|person

In this case the all pair logic would create pairs like"end stage liver disease : patient","linda : platelets","his : fluid balance" ,"she : his" which are in practice nowhere coreferent with each other, and thus here we can have some logical rules for making pairs which would restrict from creating certain pairs which are for sure not related to each other, which may need some existing knowledge base to come up with rules

Logical Rules for restricting non-related pairs from generating:

Knowledge Set: Out of all concept types"Pronoun" is a limited category and there are limited amount of pronouns in english language which can be listed on down on sheet of paper as well. Knowledge base for all the pronouns can easily be created to tag pronouns with some of the properties of the pronouns. Properties of pronouns could be listed as"Gender" (male,female,unknown) ,"Plurality" (Singular,Plural,unknown), Personal Pronouns "Personal","Non-Personal","Unknown")

Pronouns	Gender	Plurality	Personal
I		Singular	Personal
you			Personal
we		Plural	Personal
they		Plural	Personal
me		Singular	Personal
you			Personal
him	Male	Singular	Personal
her	Female	Singular	Personal
us		Plural	Personal
It			Non-Personal
This		Singular	Non-Personal
That		Singular	Non-Personal
these		Plural	Non-Personal
those		Plural	Non-Personal
itself		Singular	Non-Personal
Anything		Singular	Non-Personal

Figure 2: Pronoun Knowledge Base

Concept type of the phrases plays an important role in generating pairs as it is the common parameter which classifies the different phrases, this same basis has been used in generating pairs

4.3.2 Gender tagging for "Person" and "Pronoun" concept type

As seen in the details about concepts the "Person" concept type includes phrases related to person including the personal pronouns eg. He, She, Lynda, John. Pairing up all the phrases in "Person" type with "Person" types would result in some of the pairs which may differ in gender or their plurality which is clear indication of it being a non related pairs.

Eg.
She|15:2 15:2|person
Lynda|14:1 14:1|person
doctor|17:1 17:1|person
He|20:1 20:1|person
John|23:5 23:5|person

Here He : Lynda, John : She doesn"t sound as related pair as general knowledge help us know that John has to be male person and thus it can not be linked to a female concept. And we can now come up with rule by deducing the gender

of particular mention of type concept and avoid pairing it with mention which has opposite gender.

We use a specific library to guess the gender of person"s name called as Gender Guesser which return the gender on passing the name.Though this can is based on huge databases of names countrywise and may return "andy"/"unknown" in case a particular name androgynous, (can be used for both genders) or not present in the database, in such cases pairing could be done with both male and female concepts but this approach certainly reduces the pairs which are clearly of different genders.

Eg. The gender guesser tool reports names as "Pauley" to be androgynous and it gets mapped with both male and female mentions, such cases may not be mapped correctly in the final set of coreferent pairs as well and thus leaves up scope for improvement.

What can be done in future to address these cases?
We can look into other nearest mentions and can predict higher chances of it having the same gender as the nearest mention of personal pronouns.
Eg. Pauley suffers headache regularly, She reported it yesterday

4.3.3 Plurality tagging for Person and Pronoun category

Pronoun being a very limited set, the mentions from pronoun type can be tagged by looking up into the knowledge base for pronouns and they are mapped as singular or plural some cases such as "You" which is unmarked plurals (used as both singular and plural) is mapped as both.

For the person category a tool called as inflect engine helps us generating plurals or singulars for a name or word, using the tool mentions are marked as either "singular", "plural" or "unknown".

The tool returns singular/plural form of a word passed to it, using the conditional statements by comparing the singular and plural form of the word itself it can be deduced the plurality of the word

Pairs are then generated by avoiding the pairs for different plurality and pairing up the unknown ones with both other singular and plural mentions The concept type "Pronoun" does include certain personal pronouns, the pronoun knowledge base is used to mark the pronouns from the text with the gender, plurality and if its personal and non-personal. And all the personal pronouns pair up with the mentions from the person based on the gender and avoids pairing up with mention of different genderBased on the knowledge base used for pronouns each mention from pronoun concept type is mapped it with "personal" and "non-personal" tag Based on the above three techniques, a strong criterion can be formed to avoid certain pair makings Based on the tags for the "Person" and "Pronoun" mentions following restrictions can be put on the person : person pairs

- Male should not be mapped with female mention
- Singular mention should not be mapped with plural mention
- If gender is not present then it can be mapped with both male and female mentions
- If plurality is not present then it can be mapped with both singular and plural mentions

While matching Person with pronouns only extra constraint has been put which considers only the personal pronouns in the pairs In case of Pronoun : Pronoun Pairs along with first 4 restrictions , personal pronoun should not be paired with Non-personal one.

As Treatment, Tests, Problems and Time can be referred using pronouns, all the Non-Personal pronouns can be mapped to these other concepts

Eg.
hepatitis C cirrhosis|12:16 12:18|problem
fluid balance|40:2 40:4|test
it|39:24 39:24|pronoun

Here all other concepts like Treatment, Problem, Test, Time could be referred by a non-personal pronouns

4.3.4 Generating pairs within the concepts

Phrases within the same concept have very high chances of being co-related with each other hence for Treatment, Test , Problem and Time concept type pairs are made within the same concepts

Eg.
red cells|42:15 42:16|treatment
platelets|42:13 42:13|treatment

As above concepts fall under the same concept type "treatment" they are highly likely to be co-referent hence these will be paired up.

Challenge Faced :

Despite having the above-mentioned restriction the process may generate high number of pairs which may have large number of non-related pairs in the large sized documents.

To reduce the number of pairs certain assumptions can be made without compromising much on the accuracy, we assume that the coreferential pairs appear within a window of certain lines (line threshold)

Eg. Its highly uncertain that word from 1st line being related with word from 50th line, and we can avoid such pairings and consider pairs within certain range.

If the threshold is set to be 15 then all the phrases will be paired up which has difference between the end lines less than equal to 15

platelets|52:24 52:24|Neoral|62:13 62:13|3

platelets|52:24 52:24|CellCept|62:16 62:16|3

platelets|52:24 52:24|physical therapy|64:8 64:9|3

These assumptions would result in breaking up the longer coreferential chains into smaller ones. While training the model to classification of the generated pairs is performed after comparing with True Coreferential Pairs provided with i2b2 dataset and pairs are marked as 1 if it matches with the true data set and 0 otherwise.

Challenge Faced :

If the output after classification may result biased data with more number of unmatched pairs

To dilute the count of unmatched pairs, only the pairs falling within a window of shorter line threshold could be considered. These modifications will help getting the matching pairs from broader range.

Summary about how many pairs were generated for document and ration of true and false pairs

Challenge Faced : CLAMP tool generates multiple phrases which are substring of an existing phrase and pairs eg. severe heart disease |123:2 123:4| problem disease|123:4 123:4|problem here the "disease" is a part of larger phrase " severe heart disease"

Such concepts can be removed as it generates different pairs from the part of same phrase.

4.3.5 Extracting Features

For every mention pair a feature vector was built using properties of the two words. This feature vector would represent the mention pair and was used as the input to the classifier for finding coreferent pairs. The following features were used for representing the two concepts:

Concept type of the mention pair:

"pronoun": 0, "person": 1, "treatment" : 2, "problem":3,"test":4

Length of the common part among the two strings: The background material we surveyed used a feature that took in account the length of the longest common substring in the two concepts as a feature. But there is a possibility that there is a common subsequence of strings that may or may not occur one after the other.

For example:

c1="The cyst in her kidney"

c2="cyst in kidney"

Here if we calculate the length of longest common substring then it will be 2("cyst in"), but there are more words that are common. In c1 and c2 "the cyst in kidney" is the longest common subsequence of strings. Hence it is more likely that these concepts are coreferent if the length this feature is large

We have also considered the length of the common substring between the two concepts as a feature. The larger the length of the substring, greater the likelihood of the two concepts being coreferent.

Further another feature analyses how much both string differ from each other by subtracting the length of the common substring from the length of the larger substring.

Another feature checks if the first word in the concept matches. For example:

c1="heart condition"

c2="heart disease"

These two concepts are highly likely to be coreferent, and it would be confirmed even further since their starting word is the same.

Also, a features checks if the last word of concepts match. For example:

c1="asthma"

c2="chronic asthma"

These two concepts should be classified as coreferent. And it would be confirmed via this feature since their last word matches.

Length of both those concepts is also included in features. Also the number of word in both concepts is used as a feature. The line numbers and their indexes on the line are also included as features. Since if two words are on the same line or they are more likely to be coreferent. Also the position of the two words matters when it comes to being coreferent. Words very near to each other are less likely to be coreferent since in natural language. Also difference of line numbers of two concepts is considered as a feature since concepts are not likely to be coreferent if they are very far apart.

Also the number of words that are common between the two concepts is being considered as a feature. The ratio of common words between two concepts is also a feature that is considered. Also there is a feature that checks if the strings exactly match each other.

We also tried using the Unified Medical Language System(UMLS) REST API to search for similarity between concepts using exactSearch, wordSearch, approximateSearch methods. But the response time of the API was very large and since the number of examples also was very large it was taking a lot of time to load the features. Hence the features generated using these 3 methods were not used in the training for SVM

4.3.6 Support Vector Machine Classification

We are using Support Vector Machine to classify mention-pair into two types. Whether it is a valid mention pair or not. We are training on the dataset **generated in the mention-pair stage.**

The dataset consists of 235,944 mention pairs. Out of which 6433 are positive examples and 229,511 are negative examples. Our dataset is highly imbalanced which causes problem in correctly classifying examples.

The following example shows the features for a mention-pair "patient he"

1.0 | 0.0 | 7.0 | 1.0 | 1.0 | 7.0 | 2.0 | 1.0 | 1.0 | 10.0 | 1.0 | 10.0 | 1.0 | 10.0 | 40.0 | 10.0 | 40.0 | 0.0 | 0.0 | 1.0 | 0.5 | 0.0 | 1.0

[-0.48351224, -0.57429413, 0.0611791 , -0.48351224, -0.48351224, 0.0611791 , -0.39273035, -0.48351224, -0.48351224, 0.33352477, -0.48351224, 0.33352477, -0.48351224, 0.33352477, 3.05698149, 0.33352477, 3.05698149, -0.57429413, -0.57429413, -0.48351224, -0.52890319, -0.57429413, -0.48351224]

We have used the sklearn implementation of SVM. We are splitting the data into Train and test sets, with the split ratio being 67:33 with 67 percent data in Train set and 33 percent data in test set.

Since our data is highly imbalanced we tried training svm with class weight parameter as balanced.The âĂIJbalancedâĂİ mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n samples / (n classes * np.bincount(y)).

Other parameters like **kernel = rbf and gamma = auto** are kept the same only the value of C we have tried to tune.

We tried increasing the penalty parameter C of the error term which gave us following results.

We preprocessed the data so as to center to the mean and component wise scale to unit variance. This is the output for the same feature values in the above example.

We tried increasing the penalty parameter C of the error term which gave us following results.

C	Precision	Recall	F1
10	0.151	0.775	0.254
50	0.156	0.757	0.259
100	0.163	0.749	0.268
1000	0.170	0.723	0.276

Figure 3: Results for different values of penalty parameter

Increasing the penalty parameter C increases the time required to train the model The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples.

We tried finding the importance of features while training but the feature weights were available only when the svm kernel was linear. But still knowing those feature weights didnâĂŹt provide a concrete

We tried doing Random Under Sampling to re-

duce the number of negative examples, so that it Would balance the number of positive and negative examples.

Precision = 0.831 Recall = 0.760 Accuracy = 0.794

But this generated a model that would classify all the mention-pairs in a test file as positive.

Eventually we decided to train on a subset of data wherein we used only 20 percent of the data. Final Implementation parameters:

Penalty parameter C of the error term = 10.0(default) class weight = balanced Gamma = auto

These are the final evaluation metrics which we got after training. Precision = 0.25 recall = 0.42 F1 = 0.31

4.3.7 Chains Creation

The output of the model gives up co-referent pairs in the following form

P1 | Sl P1:Sw P1 | El P1:Ew P1 | P2 | Sl P2:Sw P2 | El P2:Ew P2 | Concept Type

Where P1,P2 are the phrases in the pair

Sl : Start Line number (Line Number at which the phrase starts)

Sw : Start Word number (Word Number of the line at which the phrase starts)

El : End Line number

Ew : End Word number

Eg.

"infant|17:1|17:1|He|19:0|19:0|1

Alleyne|18:0|18:0|He|19:0|19:0|1"

This example mentions that the "infant" and "He" are co-related and "Alleyne" and "He" are co-related Observing the position of the phrase "He", which is same in both the pairs.

It can be concluded "infant" is related to "Alleyne".

The approach is to find the pairs with where same instance of word appears and merge them together in the form of a chain.

The position of a phrase is unique for that particular word in the document , document might have same word appearing multiple times but each occurrence will have its own Sl:Sw and El:Ew combination.

Thus, a chain Infant->He->Alleyne would be considered in the output.

5 References

1. Hobbs JR. Resolving pronoun references. Lingua 1978;44:339e52.

2. Baldwin B. CogNIAC: high precision coreference with limited knowledge and linguistic resources. Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts. Stroudsburg, PA: Association for Computational Linguistics, 1997

3. Stoyanov V, Cardie C, Gilbert N, et al. Coreference resolution with reconcile. Proceedings of the ACL 2010 Conference Short Papers; Uppsala, Sweden. Stroudsburg, PA: Association for Computational Linguistics, 2010.

4. Recasens M, Marquez L, Sapena E, et al. SemEval-2010 Task 1: coreference resolution in multiple languages. Proceedings of the 5th International Workshop on Semantic Evaluation; Uppsala, Sweden. Stroudsburg, PA: Association for Computational Linguistics, 2010.

5. Ng V, Cardie C. Improving machine learning approaches to coreference resolution. Proceedings of the 40th Annual Meeting on ssociation for Computational Linguistics; University of Pennsylvania, USA. Stroudsburg, PA: Association for Computational Linguistics, 2002.

6. Bengston E, Roth D. Understanding the value of features for coreference resolution. Proceedings of the Conference on Empirical Methods in Natural Language Processing; Waikiki, Honolulu, Hawaii. Stroudsburg, PA: Association for Computational Linguistics, 2008.

7. Soon WM, Ng HT, Lim DCY. A machine learning approach to coreference resolution of noun phrases. Computational linguistics. Cambridge, MA: MIT Press, 2001;27

8. Versley Y, Ponzetto SP, Poesio M, et al. BART: a modular toolkit for coreference resolution. Proceedings HLT-Demonstrations âĂŹ08 Proceedings of the 46th Annual

Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session; Columbus, OH. Stroudsburg, PA: Association for Computational Linguistics, 2008.

9. Zheng J, Chapman WW, Miller TA, et al. A system for coreference resolution for the clinical narrative. J Am Med Inform Assoc. Published Online First. doi:10.1136/ amiajnl-2011-000599

10. Wang Y, Melton GB, Pakhomov S. ItâĂŹs about this and that: a description of anaphoric expressions in clinical text. AMIA Annu Symp Proc 2011;2011:1471e80

11. Rink, Bryan and Roberts, Kirk and M Harabagiu, Sanda. (2012). A supervised framework for resolving coreference in clinical records. Journal of the American Medical Informatics Association : JAMIA. 19. 875-82. 10.1136/amiajnl-2012-000810.

12. Raghunathan K, Lee H, Rangarajan S, et al. A multi-pass sieve for coreference resolution, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing; MIT Stata Center, MA. Stroudsburg, PA: Association for Computational Linguistics, 2010.

13. Weissenbacher, Davy and Sasaki, Yutaka. (2013). Which Factors Contributes to Resolving Coreference Chains with Bayesian Networks?.

14. https://ronan.collobert.com/senna/

15. https://nlp.stanford.edu/projects/glove/

16. Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural

17. Language Processing (EMNLP 2014), 12, 1532-1543.

18. stanfordnlp/GloVe - Pennington, Socher, and Manning's C implementation of the model

19. https://github.com/GradySimon/tensorflow-glove

20. https://nlp.stanford.edu/software/CRF-NER.html

21. https://github.com/philipperemy/Stanford-NER-Python

22. https://pypi.org/project/gender-guesser/

23. https://pypi.org/project/inflect/

`\usepackage[nohyperref]{acl2018}`.

> "We previously showed (Gusfield, 1997) ...""

should be avoided. Instead, use citations such as

> "Gusfield (1997) previously showed ...
> ""

References

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.