Concept for submission 3

Team - 7

Method – Gender Classification

Gender prediction 1 using Tf-idf vector

Approach: When we take the character prediction, if we have to predict, for example Character Robert Baratheon is Male/female, when we go to the GoT wiki page of Robert Baratheon title of the page, redirect anchor text, if this character name is matched with any name present in the Cbor file then it would be the wiki page of Robert Baratheon.

1. We take the page and first paragraph of that page and built an tf-idf vector for each and every paragraph, we do the same for each and every paragraph ion the page and build a tf-idf vectors based on the words in that paragraph
2. We will do all this for the training characters, if we look at the test and train character names, we can know the gender of each and every character. Now, we take these names and match them thoroughly with anchor text, redirect and titles.
3. The same thing we do with test data i.e; We will take all testing characters, if we look at the test and train character names, we can know the gender of each and every character. Now, we take these names and match them thoroughly with anchor text, redirect and titles.
4. Now, we will train tf-idf vectors from the first para and then train it by using logistic regression since it is a classification model which is used, in our case, to predict the binary output deciding whether the character is male / female.
5. Once we get all the tf-idf vectors for male and female we can see which one has more features, whether it is male/female, we also have Robert Baratheon tf-idf vector, we can predict whether it is male/female.

Evaluating

1. We will divide the data into train_data and test_data equally and then train the data by using the classifier (logistic regression) and then take that data and test it with the test_data, were we will get the predictions a M/F
2. Now by using the ground_truth we have, we will evaluate in which it gives the correct result 0/1. If(true){ returns 1} else{ returns 0}
3. We are planning to divide the train_data into two parts and retrain the data again and then test it with whole data, but since the data is more we don't think we will obtain good results, but we want to implement this approach and check ML Methods

Gender Prediction 2 using Coreference

1. We have chosen gender as a feature because we know the fact that death rate of male is more when compared to female so in such a way, we can predict easily who is dead/alive, that is the reason we are interested in doing gender prediction.
2. Character gender prediction- parse whole 5 textbooks using lucene and our query will be Character name + male pronous, Character name + female pronous.
3. We will take the top queries and based on score we can decide if the character is male or female and use this as a feature in our prediction.
4. Using co-reference that we mentioned in our methods, we would be trying how it works on gender. We are not sure if this will come up to our expectations but still giving out our best try. We could achieve a accuracy of 66% and with the help of coreference the score may have a bit of impact.

We can add our existing gender predictor and using co-reference we can list all the entity names instead of pronouns (in the paragraphs) and use entity linker so that the appropriate page names are being referred to .

Coreference

We are using the Stanford parser to implement this as a method. Each paragraphs/ sentences are parsed accordingly and coreference names are replaced with original entity names. We find all the expressions that refer to the same entity This method is then used in our gender predictor and allegiance predictor for parsing .

Open IE

1.In each sentence, we remove pronouns and we bring it to the subject predicate object stage. We then get relations for every entity and we put a count to it (only close relations are considered like father, brother, sister..) . For example, x has mother, father and son. So, x's relation count is 3. Later this relation count is used as a feature where we can know that if we more the relation count, his survival rate is less or more.

1. We have Open IE parser, from the Open IE paper which we read, we came to know syntactic constraint. It does a simple parts of speech tagging where it limits the relation to be either a verb (e.g.: invented) , a verb followed immediately by a preposition (e.g.: located in), or a verb followed by a nouns, adverbs or adjectives ending in a pronoun(e.g.: has atomic weight of). Using this technique, we give another feature.

Relation count

Disambiguation

Entity linking (disambiguation) – The goal is to tie together named entities to an identifying object or concept

- We make use of Game of Thrones Cbor file and did parsing {page text} parsing in the cbor file which creates mapping of all textual references in the wiki for all individual pages,
- We have identified the pages that can act as entity definition, where 'he' or 'she' from every character name is combined and we give it lucene.
- We use tf-idf methods to disambiguate. Whole para text (sentence) plus character name is given as input and we find the correct para related to that based on tf-idf score for that reference. This also is used as one feature in our prediction.
- Disambiguity is handled by using BM25, where the page name is having multiple pages referred to it, which is tried to get resolved, so BM25 is been used to get the relevant page names for an entity and map the first ranked page to this entity

We are trying to improve the disambiguation that we tried last time.

From the video link, https://www.youtube.com/watch?v=CBvE3BNErgE

We are trying to disambiguation that we could not do better last time. We identify all entity mentions in text and for each we mention we retrieve candidates. Entity is identified with surrounding text and search query now will have all those that was identified. We are using tf-idf methods to disambiguate. Eg: sansa is a disambiguation name for sansa stark entity. Sansa will have inlink from sansa stark, and anchor text.

Features

Allegiance prediction 1. –

We consider every sentence in cbor. For every sentence we find entities (character column will be our entities). We try to eliminate certain scenarios. Suppose an entity is followed by an allegiance name, most of the cases are eliminated. Next case we consider is an entity and allegiance present are same sentence. 70% of the cases get eliminated here. Our next try is nearest allegiance name near to the entity. Suppose for example, let's take 2 sentences into consideration. "x is in A house and B is also a house. A house has x and C also is there." We search through fandom and we see that entity x is followed with "A" twice in the same sentence and hence we assign entity x with allegiance A. We do this with tf-idf. We also need disambiguation for this, we need to link stansa with stansa stark. We need to remove the disambiguation name and replace it with original name.

Allegiance prediction 2 –

Bag of words model – using a vector space model, we can represent unstructured text where each dimension of vector is a specific vector/ attribute. Bag of words model represents each text document as a numeric vector where each dimension is a specific word from corpus where value indicated frequency in the document. Each column will represent words from corpus and each row represents documents. We find the count of specific allegiance names using this model and assign it to the entity.

Improve popularity-

1. Popularity – This is also used as a feature for our prediction to generate the popularity of a character i.e; whether the character is popular or not popular
    - Entity Linker is been used to create inlinks and outlinks of the graph from cbor file i.e; a character name.
    - we find the popularity score for each characters where we are trying to implement page rank algorithm.
    - Page rank is used generally to rank web pages based on search engine results. We are also trying to find the popularity score for the same.

From page rank algorithm we take inlinks, outlinks that are to be considered from our cbor file. We are trying to do better than the last time for making this feature a main feature for consideration. We give a popularity score to each character that will be used as a feature.

We have total 6 features

GP1 – Gender Prediction 1

GP2- Gender prediction 2

AP1 – Allegiance Prediction 1

AP2 – Allegiance Prediction 2

RC – Relation Count (Using Open IE)

PO- Popularity

We will be doing feature analysis and we will find which features matter, and which don't and how the F-1 scores will vary if features vary. We will try this with our Machine Learning models as before.

Machine Learning Models

1. We plan to implement minimum 6-7 machine learning methods and check the accuracy and F1- measure, for which it shows the best results
2. Following are the ML methods which we want to implement

SVM

It is based on the idea of finding a hyperplane that best separates the features into different domains. We plan to implent SVM using different kernals (linear, Radial, Polynomial) and check which performs the better

Random Forest

In a group of decision trees, the output predicted by most number of trees will be the resultant outcome, let's consider King Joffrey where random forest predicts multiple decision trees on king Joffrey Male/Female (i.e; of house Baratheon -> (Decision tree1), family -> decision tree2),.....), at last the majority of votes predicted by the decision tree becomes our model's prediction.

Logistic regression

This is a classification model which is used, in our case, to predict the binary output deciding whether the character survives or not.

F1 Logistic regression

This is a modification on logistic regression to try to maximize F1 instead of minimizing mean squared error. This method is same as logistic regression other than the objective function, where F1 equation as follows

F1 = 2 * ((Recall +Prec) / (Recall +Prec))

Neural Networks

The data is passed through a neural network where the preprocessing of data takes place in hidden layers such as feature extraction (convolution, max pooling, etc.) and the output produced in hidden layers is passed through an activation function to predict the possible binary outcome(M/F)

K-NN

We will divide data into training and test data. select a value K. determine which distance function is to be used, choose a sample from the test data that needs to be classified and compute the distance to its n training samples, sort the distances obtained and take the k-

nearest data samples, assign the test class to the class based on the majority vote of its k neighbors.

Linear regression,Bagging & Boosting

We are planning to do many methods so we will know which will give us a better score and which is more accurate. We might include more methods or if we feel too overwhelmed, we might drop one.

Contributions:

Arvind- 2 Features and one method (Allegiance and Disambiguation)

Medhini – 2 features and one method (Gender prediction using Coreference, Relation count, OpenIE)

Akhila – 2 features and one method (Gender Prediction using Tf-idf, popularity and Open IE)

As a team we have learned various concepts, different approaches and different way of solving a problem in a simple way from all the individual teammate, We have implemented and evaluated using various machine learning methods as a team so that we are able to evaluate which performs the best when compared to all other methods, Since Entity linking, openIE, disambiguity these are all the concepts which were linked together, we all had a platform to share our knowledge as a team and present it as a whole project with decent values.