Using Decision Tree Method For Classification In CR

Elov Botir Boltayevich
Computational linguistics, and Digital technologies,
Tashkent State University of Uzbek language and Literature
Tashkent, Uzbekistan
elov@navoiy-uni.uz

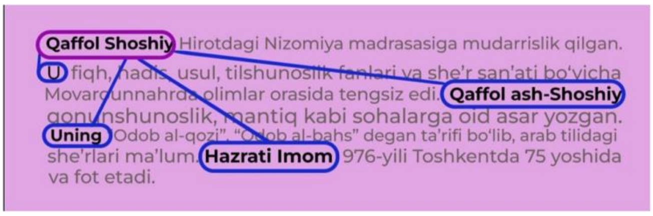
Abdisalomova Shakhlo Abdimurod kizi
Computational linguistics, and Digital technologies,
Tashkent State University of Uzbek language and Literature
Tashkent, Uzbekistan
abdisalomovashahlo@gmail.com

***Abstract* — In the realm of communication, diverse linguistic units serve to circumvent repetitive usage of identical terms. Due to its pivotal role in ensuring the coherence and logical consistency of discourse, coreference holds paramount importance in textual coherence. Coreference is a comprehensive, complex and ambiguous phenomenon. Moreover, in the contemporary era of Information Technology, the efficacy of artificial intelligence in comprehending text hinges significantly upon the precision of Coreference Resolution. Thus, In today's world, where computers managed perform almost all tasks, Coreference Resolution issues for every language has become an urgent and unavoidable task. Coreference Resolution is necessary for NLP tasks such as sentiment analysis, text segmentation, chatbots, and virtual assistants. This article is dedicated to the task of Coreference Resolution and discusses the classification phase of Coreference Resolution and the importance of the Decision Tree method in this process. It also includes the results and analysis of tests conducted on a small dataset of texts in Uzbek.**

Keywords—Coreference Resolution, Decision Tree, Classification, machine learning, algorithm, metric, method

I. Introduction

The essence of any sentence in speech is understood through the person, object, or event being discussed within the context. The human mind can directly comprehend the content of a text. However, for Artificial Intelligence to interpret it correctly, it is important to identify the mutual synonymy of text parts and the speech units connected in a single core chain that establishes coreference. This task is assigned to CR (Coreference Resolution) systems in NLP. CR is the process of automatically identifying all reference segments in a text that point to the same object. Figure 1 provides in example of this process.

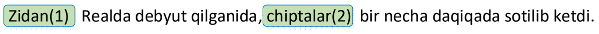


Qaffol Shoshiy Hirotdagi Nizomiya madrasasiga mudarrislik qilgan. U fiqh, hadis, usul, tilshunoslik fanlari va she'r san'ati bo'yicha Movarounnahrda olimlar orasida tengsiz edi. Qaffol ash-Shoshiy qonunshunoslik, mantiq kabi sohalarga oid asar yozgan. Uning "Odob al-qozi", "Odob al-bahs" degan ta'rifi bo'lib, arab tilidagi she'rlari ma'lum. Hazrati Imom 976-yili Toshkentda 75 yoshida va fot etadi.

Fig. 1. The phenomenon of coreference in Uzbek texts.

It should be noted that we refer to the task of coreference resolution as a general problem of identifying and resolving references in a text. However, technically, there are several types of references, and their definitions can be a subject of debate.

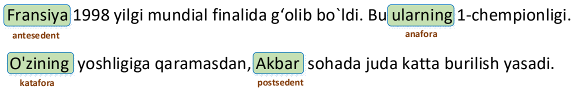
A different case from the task of coreference resolution is the issue of resolving anaphora. An anaphoric relationship arises in a text when one term refers to another, determining the interpretation of the latter. In the following example, (1) and (2) refer to different real-world objects. However, they are used in the same context, and the interpretation of (2) relies on (1). These references are not joint sentences but are connected by anaphora. We describe Anaphora Resolution in Uzbek texts as follows:



Zidan(1) Realda debyut qilganida, chiptalar(2) bir necha daqiqada sotilib ketdi.

Fig. 2. Anaphora Resolution in Uzbek texts.

Although anaphoric relationships differ from coreference, in most cases, they are equivalent. There are many examples of such distinctions and other types of references. However, Coreference Resolution has a broader scope and covers a significant portion of existing work. The main difference between these concepts is that anaphora occurs after the word it refers to in a sentence, while cataphora occurs before it. The word that comes before an anaphora is called an antecedent, while the word after a cataphora is called a postcedent. An example of this phenomenon is given below based on Uzbek texts:



Fransiya 1998 yilgi mundial finalida g'olib bo`ldi. Bu ularning 1-chempionligi.

O'zining yoshligiga qaramasdan, Akbar sohada juda katta burilish yasadi.

antesedent
anafora
katafora
postsedent

Fig. 3. Anaphora and cataphora in Uzbek texts.

In general, the tasks of resolving anaphora and coreference are stages in the semantic analysis of a text. Through them, the potential for a machine to "think" like a human can be developed. This article discusses the importance of the decision tree method in coreference resolution and the results of its application.

Initially, the issue of coreference was resolved through rule-based linguistic approaches. In today's technological age, it is being resolved through machine learning methods. Machine learning is an approach that involves training models with algorithms to perform tasks usually done by humans. Using this approach, the stages of resolving coreference are as follows:

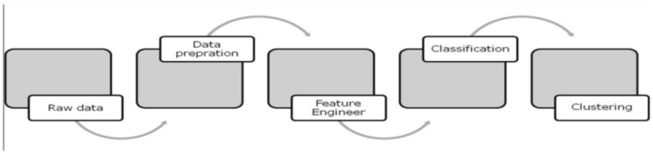


Fig. 4. Stages of resolving coreference in machine learning.

The article discusses the classification stage of resolving coreference, focusing on the significance of the Decision Tree method and the results of its application.

II. Literature Analysis

While there are various methods used for classification in the process of resolving coreference, many studies have employed the Decision Tree method. Among such works are studies by K.Šteflović, J.Kapusta [1], Soon et al. [2], Ng and Cardia [3], R.V.S. Ram and S. Lalitha Devi [4], X.Yang et al [5]. Additionally, Dinh Le Thanh's report highlights the importance of the Decision Tree method in classifying coreferent entities and provides formulas for calculating result accuracy [6]. The study by S. Pogorily and P. Biletskyi compares CNN, BiLSTM, RoBERTa, and Decision Tree methods in resolving coreference in Ukrainian text and analyzes the results [7]. J. McCarthy & W. Lehnert's article explains the Decision Tree principles in classification using the MUC-5 system, while Z. Dźunić et al. explore the MUC-6 coreference corpus in their work [8, 9]. E. Fernandes et al. tested the latent tree method in three languages, achieving a 60.15% result [10].

In many studies, the coreference problem is solved using multiple modern methods, with the Decision Tree method consistently being included. This is because, upon analyzing research utilizing the Decision Tree method, we see that the model's accuracy consistently shows positive results. Thus, despite the passage of time, the Decision Tree method remains one of the most widely used classification methods due to its effectiveness.

III. Application of the Decision Tree Method in Coreference Resolution

A. General Description of the Decision Tree method

Classification is a supervised learning method aimed at predicting the class of given data points. There are various classification algorithms. The main classification algorithms include Decision Tree, Naive Bayes, Artificial Neural Network, and K-Nearest Neighbor. The Decision Tree method is widely used in classification due to its convenience in handling categorical and continuous attributes. It has a hierarchical tree structure consisting of root nodes, branches, internal nodes, and leaf nodes. In this algorithm, data is viewed as a set of if-then rules based on feature values. The Decision Tree method has several advantages in classification, which are listed below:

- Ability to select features;
- Capability to handle discrete and continuous attributes;
- Ability to process unknown values;
- Easy modification of rule sets;
- Simplicity for understanding and interpretation;
- Robustness and ability to analyze large datasets in a short time.

The Decision Tree algorithm is built on the "top-down" principle, meaning:

– The algorithm starts with the "root node," which represents the entire dataset;

– It searches for the primary feature that answers the question, "Which attribute is the best choice for the given node?" in order to divide the data into clear groups;

– Based on the answer to this question, it splits the data into smaller subsets and creates new branches;

– The algorithm continues asking questions and divides the data into final "leaf nodes" that represent the predicted outcomes or classifications in each branch.

In the Decision Tree method, different algorithms can be used depending on the dataset. The most popular Decision Tree algorithms include ID3, C4.5, CART, and Random Forest. Figure 3 illustrates the general structure and functioning of the Decision Tree algorithm.

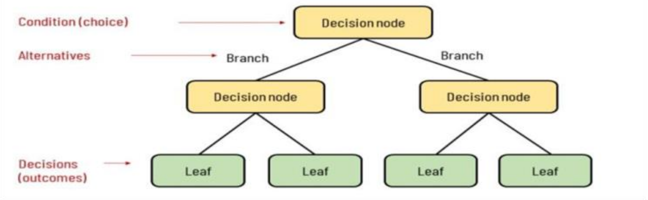


Fig. 5. Decision tree structure.

It seems that the Decision Tree method is a crucial tool for understanding the relationships between input variables and their outcomes and for identifying the most important features that help in reaching the final decision.

B. Using the Decision Tree

It is known that a coreference corpus is necessary for the model to work. In this article, we perform classification on a small dataset (6,185 tokens) composed of Uzbek language texts. Based on this dataset, we use 11 features for classifying candidate antecedents and anaphors:

- HEAD_MATCH: Do the mentions have the same head noun? Possible values: {yes, no}
- STR_MATCH: Do the mentions have the same string? Possible values: {yes, no}
- NUMBER: Are the mentions compatible in number? Possible values: {yes, no}
- GENDER: Are the mentions compatible in gender? Possible values: {yes, no, unknown}
- ANIMACY: Are the mentions compatible in animacy? Possible values: {yes, no, unknown}
- BOTH_PRONOUNS: Are both mentions pronouns? Possible values: {yes, no, unknown}
- BOTH_PROPER_NOUNS: Are both mentions proper nouns? Possible values: {yes, no, unknown}
- APPOSITIVE: Are the mentions in an appositive relationship? Possible values: {yes, no}
- SEMCLASS: Are the mentions in the same semantic class? Possible values: {yes, no, unknown}
- DISTANCE: The number of sentences separating the mentions. Possible values: {0, 1, 2, 3...}
- ALIAS: Is one mention an alias of the other? Possible values: {yes, no}
- HEAD_MATCH: Do the mentions have the same head noun? Possible values: {yes, no}
- STR_MATCH: Do the mentions have the same string? Possible values: {yes, no}
- NUMBER: Are the mentions compatible in number? Possible values: {yes, no}
- GENDER: Are the mentions compatible in gender? Possible values: {yes, no, unknown}
- ANIMACY: Are the mentions compatible in animacy? Possible values: {yes, no, unknown}

- BOTH_PRONOUNS: Are both mentions pronouns? Possible values: {yes, no, unknown}
- BOTH_PROPER_NOUNS: Are both mentions proper nouns? Possible values: {yes, no, unknown}
- APPOSITIVE: Are the mentions in an appositive relationship? Possible values: {yes, no}
- SEMCLASS: Are the mentions in the same semantic class? Possible values: {yes, no, unknown}
- DISTANCE: The number of sentences separating the mentions. Possible values: {0, 1, 2, 3...}
- ALIAS: Is one mention an alias of the other? Possible values: {yes, no}

In our dataset, object-object type connections are frequently encountered. We will analyze whether to group them into one class or not using the Decision Tree method as an example:

Shamol energiyasi

Quyosh energiyasi

The structure for determining the category of the given units is as follows:

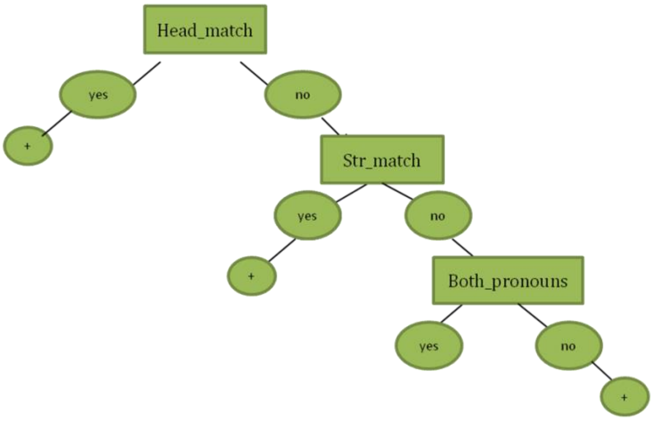


Fig. 6. Decision tree method for classification in CR.

As an example, we described the working process of a Decision Tree using 3 features. In it, positive values are more frequent than negative ones. However, when positive and negative values are equal, the process of grouping them into a class becomes more difficult. Therefore, in such cases, Entropy is used. Entropy is nothing but the uncertainty in our dataset or measure of disorder. The formula for Entropy is shown below [12]:

$$E\ (S) = -p_{(+)} \log p_{(+)} - p_{(-)} \log p_{(-)}$$

Here,

- p_+ is the probability of positive class
- p_- is the probability of negative class
- S is the subset the training example

Information gain measures the reduction of uncertainty given some feature and it is also a deciding factor for which attribute should be selected as a decision node or root node.

$$Information\ Gain = E(Y) - E\ (Y/X)$$

Using these formulas, the most appropriate attributes for decision-making are selected. As a result, the model accuracy reflects a positive value.

IV. Evaluation

In the classification type of machine learning, there are several metrics to check how accurate the models are. In the article, we perform classification on a small dataset (6,185 tokens) composed of Uzbek language texts. We use the MUC metric to evaluate this process. The MUC metric calculates how many correct connections the model has found. For the evaluation of this process, we choose the F1-Score metric, as it provides more accurate results when the ratio of positive and negative values is imbalanced. In measuring the F1-Score, the Precision and Recall parameters are first determined [11]:

Precision shows how many of the connections found by the model are actually correct.

$$Precision = \frac{Correct\ connections\ found\ by\ the\ system}{Total\ connections\ found\ by\ the\ system}$$

Recall – identifies the correct connections found by the model.

$$Recall = \frac{Correct\ connections\ found\ by\ the\ system}{Correct\ connections\ based\ on\ the\ gold\ standard}$$

The F1-Score is a metric expressed through Precision and Recall, and its formula is as follows:

$$F1\ score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

The algorithm for Precision and Recall accuracy is provided below:

```
fromsklearn.metrics.cluster import precision_score, recall_score
# Gold standard connections and model results
gold_clusters = [[1, 2, 3], [4, 5], [6]]
pred_clusters = [[1, 2], [3, 4, 5], [6]]
defmuc_score (gold, pred):
correct_links = 0
total_links = 0
for g in gold:
correct_links += len(g) - 1  # correct connections based on the Gold standard
for p in pred:
total_links += len(p) - 1  # connections found by the model
muc_recall = correct_links / total_links
muc_precision = correct_links / total_links
returnmuc_recall, muc_precision
```

Fig. 7. Algorithm for calculating Precision and Recall.

Based on the results of the calculations above, the accuracy on the small dataset of Uzbek texts is presented in Table 1 below:

TABLE 1.Accuracy of Calculating in Uzbek Texts

MUC	
Precision	73.53%
Recall	29.07%

F1	41.64%

From the analysis of the results, it can be seen that the model has a negative accuracy score. Nevertheless, the model performed effectively for Precision. The low accuracy of the model is related to the small size of our dataset. The larger the dataset, the higher the model's accuracy will be. This is a result of a small experiment, and in our future research, we will strive to address these shortcomings.

V. CONCLUSION

This article discussed the significance of the Decision Tree method and its application results in Resolving the issue of Coreference. Since the experiment was conducted on a small dataset of Uzbek texts, the model did not yield the expected results. This opens the way to performing the test with other evaluation metrics and conducting a comparative analysis of the results. In many cases, the Coreference issue is solved not by one, but by several modern methods. Among them, the Decision Tree method is always present. When we analyze studies where the Decision Tree method was applied, we see that in all cases the model accuracy had a positive result. Thus, despite the passage of time, in terms of efficiency, the Decision Tree method remains one of the most widely used classification methods.

REFERENCES

[1] K.Šteflović, J.Kapusta, Coreference Resolution for Improving Performance Measures of Classification Tasks, Applied Sciences, 2023, 13, 9272. https://doi.org/10.3390/app13169272, pp.1-20.

[2] W. Soon, H. Ng, and D. Lim (2001), A Machine Learning Approach to Coreference Resolution of Noun Phrases, Computational Linguistics 27 (4), pp. 521–544.

[3] V. Ng, C. Cardie, Improving machine learning approaches to coreference resolution, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 104-111, 2002.

[4] V. S. Ram, S. L. Devi, Clause Boundary Identification Using Conditional Random Fields, A. Gelbukh (Ed.): CICLing 2008, LNCS 4919, pp.140-150, 2008.

[5] X. Yang, J. Su, G. Zhou, Ch. L. Tan, An NP-Cluster Based Approach to Coreference Resolution, Proceedings of the 20th international conference on Computational Linguistics, 2004, pp.226-es. https://doi.org/10.3115/1220355.1220388

[6] D. Le Thanh, Two machine learning approaches to Coreference Resolution, PFL054 2008/09.

[7] S. Pogorily, P. Biletskyi, Analysis of Decision Trees for Coreference Resolution Task Ukrainian Language, CEUR-WS.org, Vol-3646, p. 255-262.

[8] J. F. McCarthy, W. G. Lehnert, Using Decision Trees for Coreference Resolution, To appear in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, V.1, 1995.

[9] Z. Dźunić, S.Momčilović, B.Todorović, M.Stanković, Coreference Resolution Using Decision Trees, 2006 8th Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Serbia, 2006, pp. 109-114, doi: 10.1109/NEURAL.2006.341188.

[10] E. R. Fernandes, C. N. dos Santos, R. L. Milidiu, Latent Trees for Coreference Resolution, Association for Computational Linguistics, 2014. doi:10.1162/COLLa.00200.

[11] Accuracy, Precision, Recall and F1 Score: Interpretation of Performance Measures. URL: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#

[12] What is Decision tree_[A step-by-step-guide] www.geeksforgeeks.org/decision-tree