IMPROVE THE EFFICIENCY TO SEARCH FOR VIETNAMESE INFORMATION WITH COREFERENCE RESOLUTION AND EVENT-ORIENTED SEMANTIC MODEL OF TEXT

[a]LE DINH SON, [b]TRAN VAN AN

[a,b]Le Quy Don Technical University, 84.24, 236 Hoang Quoc Viet, Hanoi, Vietnam
email: [a]sonld@lqdtu.edu.vn, [b]tavistu@gmail.com

Abstract: In this article, we present a coreference resolution using an event-oriented semantic model of text to search and classify text by the content from a set of documents. We have developed a test of the coreference dataset as a basis for improving search functionality based on a set of synonymous queries and indexing the content of obtained results based on the event-oriented semantic model of text. The article also proposes a mathematical model for indexing calculation based on the semantic relation of Vietnamese texts and some entities with English names. The article presents in detail the process of indexing systems such as pre-processing steps, using coreference dataset, extracting and indexing documents according to the semantic model of text.

Keywords: Coreference resolution, Vector space model, Semantic model of text, Entities with specific names.

1 Introduction

Nowadays, searching for information on the Internet has become an urgent need for most users, but we often face difficulties when sources of information are duplicated. For example, in the field of sports, when an event takes place, many online newspapers will produce information, the user's concern is to read the most accurate and complete information about the event without spending much time reading from different sources. Currently, websites allow users to search and classify for specific purposes (which may be commercial), which makes it difficult for users.

For the above reasons, we propose a process of indexing of information sources to provide the most complete and accurate content for readers. We apply methods of word separation, labeling in the preprocessing step of the Vietnamese language, build a coreference dataset and build an event-oriented semantic model of text, from which to study and propose methods of indexing texts in order to set up an indexing system of text in information systems.

The system is based on using a coreference resolution to improve the searching results based on a set of synonymous queries instead of using only the original query. This greatly increases the searching efficiency with texts whose content is closely related to the user's intent, making the search for semantics improved. For that reason, the system produces better semantic searching results than using only the original query string on some search engines.

We have developed a test system to evaluate the results of the application of the above-proposed methods, including the application of a coreference resolution in the formation of synonymous query sets, combining with the use of the semantic model of text and indexing algorithms based on that model, and tests with actual data.

1.1 Model of System

The original query will be pre-processed, then use a coreference resolution to produce a set of synonymous queries, which will be put into search engines in turn to improve search efficiency. The results of the content of the websites will be saved and the semantic model of text will be applied to the index.

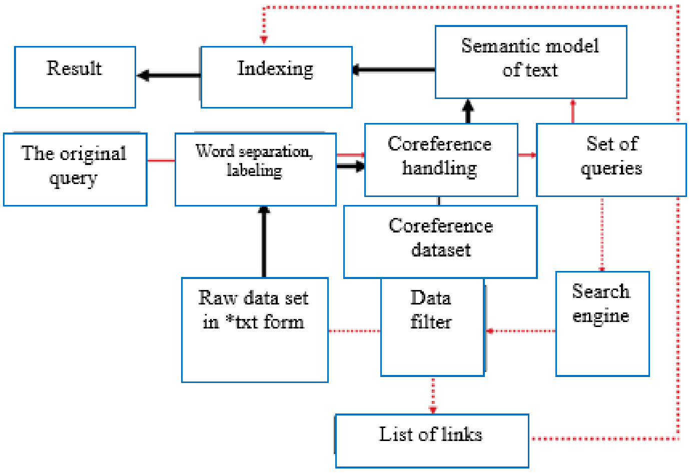


Figure 1. System Model

1.2 Pre-processing Texts: Word Separation, Labeling

We use two pre-processing modules including:

Word separation module to separate words in the text.

Labeling module to label from the category after the word has been separated.

Both of these two modules are used in the pre-processing step of the input text. However, it stems from the complexity of natural language, the accuracy in word separation and labeling has not reached the absolute level, which affects the results with the problems using word separating, labeling. Therefore, the coreference resolution uses a rule, the author has combined with training data to separate words and re-label specific names to increase the accuracy of word separation and labeling.

1.3 Coreference Resolution in Text

The problem of determining the coreference in text is the problem of determining the phrases in a same document referring to a defined entity in the real world and clustering these words into coreference series. (1-3) This is a difficult issue of processing natural language. For the Vietnamese language, this problem still poses many challenges due to its complexity and inadequate language resources. However, it is a problem with high potential of exploitation for Vietnamese data sources, which should be explored and researched.

The concept of coreference relation by Véronique Hoste (4): "The relation of coreference is the relation between two or more phrases that refer to a specific entity in the real world."

To further clarify the concept of coreference, we consider the following example: "Rooney is the captain of Man United. He is the soul of the Red Devil".

In the above example, the pronoun "He" and the noun "Rooney" have a coreference relation because they refer to a human entity named "Rooney". The words "Man United" and "Red Devil" refer to the same entity which is "Manchester United Football Club ".

With the model of solving the coreference in text, the authors in the study stated the problem of coreference in text as follows: "Solving the coreference in text is a problem defining phrases (nouns or pronouns) in a document with coreference relation and collecting these phrases into coreference strings." (5, 6) With the above statement, the input and output data of the problem can be determined as follows:

Input: Natural language text

Output: Groups of coreference words

In this research, the author uses the method of determining the coreference groups based on the rules and the coreference dataset that are built and classified manually. The coreference resolution algorithm is presented as follows: After labeling the input data, conduct the matching with the database of coreference samples, which is in the statistic data and groups, entity representing the coreference string is the first entity of each data line, corresponding to the highest statistical value (most commonly used). The next step, the entities in each document will be replaced by representative coreference entities, which serves the evaluation of the classification later. The result of a coreference group will have a structure as the following example:
Liverpool / Np: 428 | The_Kop / Np: 11 | Home_team_Anfield/Np: 2.

The above example shows the nouns and corresponding occurrences in the texts. Coreferences such as The_Kop / Np, Home_team_Anfield / Nps will be replaced by the most listed entity representing the coreference group that is Liverpool / Np.

To detect the coreference in documents needing to be reviewed, we shall develop the following algorithm:

```
Call $t_i$ - the entity in text and $f$- input text data. Function $match$ $(t_i,f)$ - the function that checks the appearance of entity $t_i$- in f.
Approve $t_i$ in the text $d_j$
If match ($t_i$, f) – then replace ($t_i$, representative coreference ($t_i$))
Next $t_{i+1}$.
```

1.4 Building the Event-oriented Semantic Model of Text

The semantic model of text often relies on semantic relations between concepts. (7) Establishing the relation between concepts will increase the semantics of sentences or paragraphs. Through semantic relations, the search results will be interlinked, which means that when searching for information, in addition to the exact results returned, it is possible to get the results that are semantically related to those results. To do that, it is necessary to build a suitable model of semantic text presentation, structuring the text as well as finding and quantifying the relations between the elements in the text.

With the above idea, the study has proposed a semantic model of text as well as the processing of text from raw structure to the following structure (8):

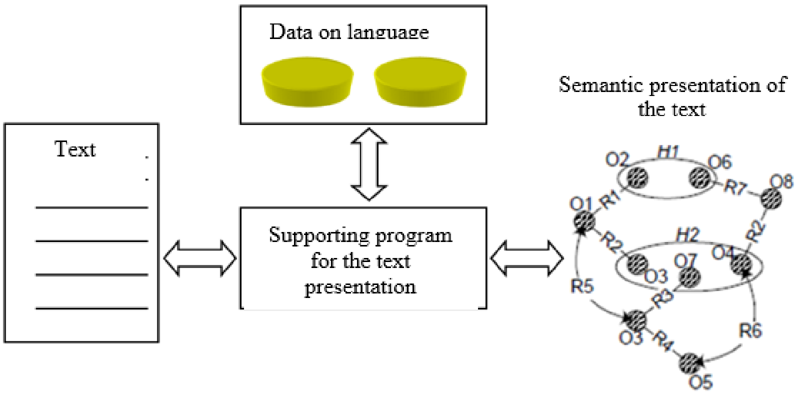

Data on language
Text
Supporting program for the text presentation
Semantic presentation of the text
O2
H1
O6
O8
R7
O1
R1
R2
H2
O7
O4
R2
R5
O3
R3
O3
R4
R6
O5

Figure 2. The Semantic Model of Text

The data of the input texts is processed by a program supporting the semantic text representation through language databases, thereby extracting relations between entities and representing them in the form of semantic graphs. In which:

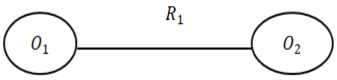
O_i - Concepts, they are nouns referring to entities,

R_i- Relation between objects.

Semantic representation of text is created from semantic representation of separate sentences of the text, their elements are concepts extracted from the analyzed texts and there are semantic relations between them. (9) Text semantic representation is expressed in graphs, each vertex of the graph is concepts, each edge is semantic relations between them. We can take an example from the following sentence:

"The wind blows the leaf. The wind passes through the gap"

With the sentence: "The wind blows the leaf", the semantic relation is expressed as follows:



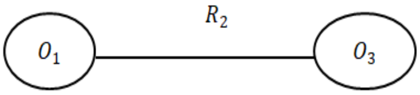
In which:

$$R_1 = \text{blow}(O_{1,}O_2)$$

O_1- "the wind"

O_2- "the leaf"

The semantic relation in the second sentence "The wind passes through the gap" as follows

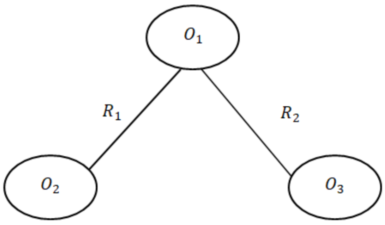


$R_2 = \text{pass through}(O_1, O_3)$

O_1- “The wind”

O_3- “the gap”

For the above two sentences in the same text, we shall have the following relation:



The above semantic model of text still has some drawbacks. Firstly, when the amount of data is large, the graph will rapidly increase in the number of vertices and edges, which greatly affects the performance of storing and querying information. Secondly, events stand aloneand not highly practical in the application of information search. For events with many entities participating in, or the entities with the same coreference, the above model has not integrated them together yet. This is shown in the following example: “MU and Man City entered to play theManchester derby match this weekend. The Red Devil is being evaluated higher than MCFC” (1). With the above model, it will be difficult to build a semantic model of text due to the following: the first entity includes both" MU "and" Man City", the semantic relation between these two entities is two verbs:"enter","play", the second entity is"the Manchester derby match". For the second sentence, the first entity is "Red Devil", the second entity is "MCFC" with the semantic relation that is "evaluated". The example above has semantic relations as follows:

$R_1 = \text{enter } (O_1, O_3)$, with O_1- “MU”, O_3- “the Manchester derby match”

$R_2 = \text{enter}(O_2, O_3)$, with$O_2$- “ Man City”, O_3- “the Manchester derby match”

$R_3 = \text{play } (O_1, O_3)$

$R_4 = \text{play } (O_2, O_3)$

$R_5 = \text{evaluated } (O_4, O_5)$, with O_4- “ Red devil”, O_5- “MCFC”

Through the semantic expression of the text as above, an original text will produce many relations with which these representations also partly lose the original meaning of the sentence, making the semantics partlychanged by the order of elements in the sentence are not represented in the model.

In search queries, users often pay attention to nouns and verbs, sentences containing nouns and verbs which are meaningful are considered an event. In other words, an event is created by arranging nouns and verbs provided that the arrangement is meaningful. Through modeling the text into a model of events, finding data based on text will return more semantic results with the query. For these reasons, the author proposes an event-oriented semantic model of text as follows:

$M =< O_t, V_t > \ (2)$

O_t- noun phrase

V_t- verb phrase

For this model, a sentence or a text can be represented as an ordered set of O_t andV_t. So a text can be represented in the following form:

$O_1 - V_1 - O_2$

In general, a text can be represented in the following form:

$O_1 \dots O_n - V_1 \dots V_m - O_k \dots O_h$ (3)

With$n, m, k, h \geq 0$.

Using event-oriented semantic representation of text, combining with co-reference, the sentence (1) shall be represented as follows

$O_1 O_2 - V_1 V_2 - O_3$ và $O_1 - V_3 - O_2$

In which

O_1- “MU” (coreference with Red Devil),

O_2- “ Man City” (coreference with MCFC),

O_3- “ the Manchester derby match”

V_1- “ enter”,

V_2- “ play”,

V_3- “evaluated”.

In this semantic model of text, the semantic representation will become simpler and semantically guaranteed. In the Information Retrieval (IR) systems, this model also helps improve the capable of querying documents based on words or analyzing phrases of the content and it produces more accurate ranking results. (10)
Applying the above model to the ranking of search results will help the search engines display the most semantically relevant results at the top of the list, the results will appear according to semantic priority compared with the input queries. In particular, the queries as well as the searched text are modeled by an event-oriented semantic model of text. With the above proposal, the queries are modeled: $Q =< O_t, V_t >$.

Considering the specific case with the following query: "MU borrows Falcao", the event-oriented semantic model of text of this query will be $Q = (O_1, V_1, O_2)$. WithO_1- “MU”, V_1- “borrow”, O_2- “Falcao”. It is possible to model the above query as a vector with the corresponding value of $q = [1,1,1]$with three dimensions of O_1, V_1, O_2. Queries can be generated from the original query is as follows:

Type of query with one missing component:

$Q_{11} = [0,1,1]$ (missing O_1), $Q_{12} = [1,0,1]$ (missing V_1), $Q_{13} = [1,1,0]$ (missing O_2)

Type of query with two missing components:

$Q_{212} = [0,0,1]$ (missing O_1, V_1) ,
$Q_{213} = [0,1,0]$ (missing O_1, O_2) ,
$Q_{223} = [1,0,0]$ (missingV_1, O_2)

So, with the above query, it is possible to create seven near-meaning queries (including the original query), in which the semantics at the highest priority is the original query, the next is the one with one and two missing components. Semantics are reduced in the incremental direction of missing elements in the query.

In the general case, we can consider $Q =< O_t, V_t >$, with the direction of the query vector as$|Q| = |O_t| + |V_t| = n$. We have the vector of the original query:$Q = [q_1, q_2, \dots q_n]$ with$q_i = 1\ (i = \overline{1,n})$

After that, we consider the cases of incomplete semantics as the query but they still take the context of the query by phasing out entities from n entities to 1 entity. Therefore, we have:

At the first priority, compared to the original query, the number of missing entities in this level is 1. We assume that $Q_{1j}$is the jth vector at level 1. That vector is represented as follows:

$$Q_{1j} = [q_1, q_2, \dots q_n] \text{ with}$$
$$\text{q} = 1\ (i = \overline{1,n},\ i <> j), q_j = 0$$

It can be seen that, at level 1, the number of queries is C_n^1

Also with the above representation, the vector of the queries at the second priority is as follows

$$Q_{2jk} = [q_1, q_2, \dots q_n] \text{ with}$$
$$q_i = 1\ (i = \overline{1,n},\ i <> j, k;\ \ j <> k), q_j = 0, q_k = 0$$

Similarly, the number of queries at level 2 is C_n^2

Present queries and the number of queries at priority levels are similarly as calculation at level 1 and 2. Number of queries at level i is C_n^i

Include the original query, the number of priority levels from 0 (priority level of the original query) to $n - 1$ (n priority level). The total number of queries counted is:

$$N = 1 + C_n^1 + C_n^2 + \dots C_n^i + \dots + C_n^{n-1}$$

with n is the direction of the query vector.

The query search process in the text set D is implemented after the queries at those levels have been identified. Call d_i as the ith document in the document set D, perform a search for the occurrence of queries at priority levels in each document d_i, the result obtained is the n-dimensional vector, corresponding to n priority levels. The vector symbol is V(d_i)

$$V(d_i) = (v_{i0}, v_{i1}, \dots, v_{i(n-1)})$$

In which$v_{ij}$is the number of occurrences of the queries at the jth priority level in the text d_i.

The efficiency of searching for information is a recommendation for users but the most concise and accurate information according to the query is included. Therefore, the authors propose a method of indexing texts by calculating the scores of the query results. In other words, from the vector $V(d_i)$, we calculate the scores for each of those vectors and then arrange the documents into the list in order from high to low scores. The function “Scores”of vector $V(d_i)$ is proposed by the authors as follows:

$$\text{Scores}(d_i) = \sum_{j=0}^{n-1} \frac{v_{ij}}{(j+1) * S} \quad (4)$$

With

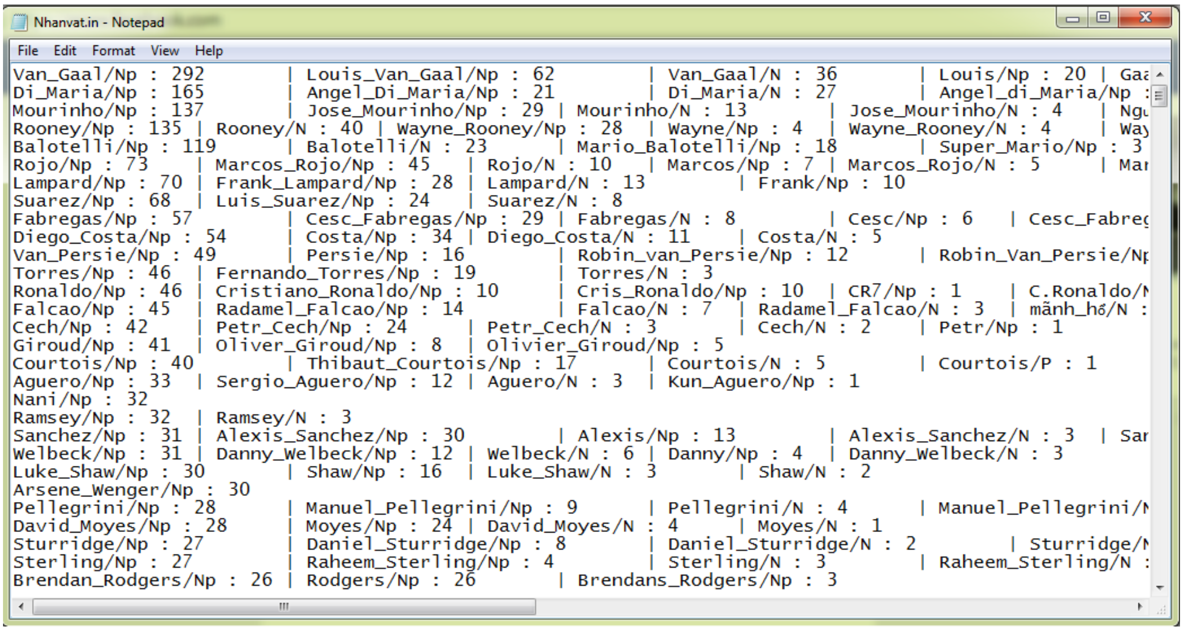
$$S = \begin{cases} \max(v_{ik}), (i, k = \overline{1,n}), & \text{if } \max(v_{ik}) \neq 0 \\ & 1, \text{if } \max(v_{ik}) = 0 \end{cases}$$

With the “Scores” function as above, according to the input query string, each document$d$in the document set D will have its own Score$s(d)$ value, the larger the value is, the greaterthe semantic relevance between the text and the query is.

2 Applying and Testing

Based on the research results, the authors have built a program to index news based on the queries. The data set was built by getting information from 2500 sports articles. The program has also carried out pre-processing steps such as word separation, labeling, entity identification and replacement of coreferences, and then modeled semantic relations and classified.

Applying the ideas of the study, the authors have built a coreference dataset based on semantic models. The structure of each file is a set of lines, each of which is a set of coreferences and the frequency of occurances, the biggest frequency is placed at the beginning of each line:



```
Van_Gaal/Np : 292       | Louis_Van_Gaal/Np : 62        | Van_Gaal/N : 36       | Louis/Np : 20 | Gaa
Di_Maria/Np : 165       | Angel_Di_Maria/Np : 21        | Di_Maria/N : 27       | Angel_di_Maria/Np :
Mourinho/Np : 137       | Jose_Mourinho/Np : 29 | Mourinho/N : 13       | Jose_Mourinho/N : 4   | Ngu
Rooney/Np : 135 | Rooney/N : 40 | Wayne_Rooney/Np : 28  | Wayne/Np : 4  | Wayne_Rooney/N : 4    | Way
Balotelli/Np : 119      | Balotelli/N : 23      | Mario_Balotelli/Np : 18       | Super_Mario/Np : 3
Rojo/Np : 73    | Marcos_Rojo/Np : 45   | Rojo/N : 10   | Marcos/Np : 7 | Marcos_Rojo/N : 5     | Mar
Lampard/Np : 70 | Frank_Lampard/Np : 28 | Lampard/N : 13        | Frank/Np : 10
Suarez/Np : 68  | Luis_Suarez/Np : 24   | Suarez/N : 8
Fabregas/Np : 57        | Cesc_Fabregas/Np : 29 | Fabregas/N : 8        | Cesc/Np : 6   | Cesc_Fabreg
Diego_Costa/Np : 54     | Costa/Np : 34 | Diego_Costa/N : 11    | Costa/N : 5
Van_Persie/Np : 49      | Persie/Np : 16        | Robin_van_Persie/Np : 12      | Robin_Van_Persie/Np
Torres/Np : 46  | Fernando_Torres/Np : 19       | Torres/N : 3
Ronaldo/Np : 46 | Cristiano_Ronaldo/Np : 10     | Cris_Ronaldo/Np : 10  | CR7/Np : 1    | C.Ronaldo/N
Falcao/Np : 45  | Radamel_Falcao/Np : 14        | Falcao/N : 7  | Radamel_Falcao/N : 3  | mãnh_hổ/N :
Cech/Np : 42    | Petr_Cech/Np : 24     | Petr_Cech/N : 3       | Cech/N : 2    | Petr/Np : 1
Giroud/Np : 41  | Oliver_Giroud/Np : 8  | Olivier_Giroud/Np : 5
Courtois/Np : 40        | Thibaut_Courtois/Np : 17      | Courtois/N : 5        | Courtois/P : 1
Aguero/Np : 33  | Sergio_Aguero/Np : 12 | Aguero/N : 3  | Kun_Aguero/Np : 1
Nani/Np : 32
Ramsey/Np : 32  | Ramsey/N : 3
Sanchez/Np : 31 | Alexis_Sanchez/Np : 30        | Alexis/Np : 13        | Alexis_Sanchez/N : 3  | Sar
Welbeck/Np : 31 | Danny_Welbeck/Np : 12 | Welbeck/N : 6 | Danny/Np : 4  | Danny_Welbeck/N : 3
Luke_Shaw/Np : 30       | Shaw/Np : 16  | Luke_Shaw/N : 3       | Shaw/N : 2
Arsene_Wenger/Np : 30
Pellegrini/Np : 28      | Manuel_Pellegrini/Np : 9      | Pellegrini/N : 4      | Manuel_Pellegrini/N
David_Moyes/Np : 28     | Moyes/Np : 24 | David_Moyes/N : 4     | Moyes/N : 1
Sturridge/Np : 27       | Daniel_Sturridge/Np : 8       | Daniel_Sturridge/N : 2        | Sturridge/N
Sterling/Np : 27        | Raheem_Sterling/Np : 4        | Sterling/N : 3        | Raheem_Sterling/N :
Brendan_Rodgers/Np : 26 | Rodgers/Np : 26       | Brendans_Rodgers/Np : 3
```

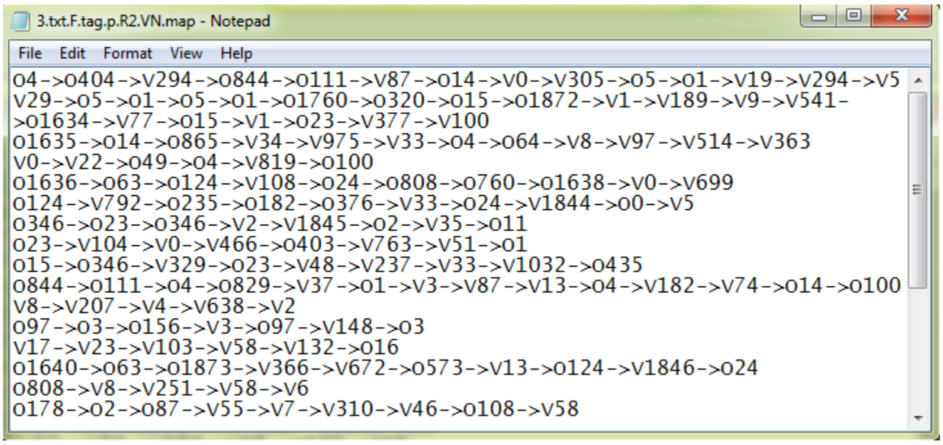
Figure 3. The Structure of a Coreference Dataset File

For texts, after having been extracted, the entity shall be separated, and each sentence is built into event-oriented semantic models as in the model (2). Example: Sentence: "Man United borrowed Ramadel Falcao in this season."

After labeling, extracting the entity, replacing the co-reference, it will become: Man_United / Np borrow/ V/ V Falcao / NP. The semantic models are represented as follows:

O1->V3->V2->O23:

In practice, the authors performed the semantics of semantic texts as the image below:



```
o4->o404->v294->o844->o111->v87->o14->v0->v305->o5->o1->v19->v294->v5
v29->o5->o1->o5->o1->o1760->o320->o15->o1872->v1->v189->v9->v541-
>o1634->v77->o15->v1->o23->v377->v100
o1635->o14->o865->v34->v975->v33->o4->o64->v8->v97->v514->v363
v0->v22->o49->o4->v819->o100
o1636->o63->o124->v108->o24->o808->o760->o1638->v0->v699
o124->v792->o235->o182->o376->v33->o24->v1844->o0->v5
o346->o23->o346->v2->v1845->o2->v35->o11
o23->v104->v0->v466->o403->v763->v51->o1
o15->o346->v329->o23->v48->v237->v33->v1032->o435
o844->o111->o4->o829->v37->o1->v3->v87->v13->o4->v182->v74->o14->o100
v8->v207->v4->v638->v2
o97->o3->o156->v3->o97->v148->o3
v17->v23->v103->v58->v132->o16
o1640->o63->o1873->v366->v672->o573->v13->o124->v1846->o24
o808->v8->v251->v58->v6
o178->o2->o87->v55->v7->v310->v46->o108->v58
```

Figure 4. A Text File After Having Been Semantically Modeled

With a dataset which is semantically modeled in an event-oriented manner, the program conducts document indexing based on the semantic matching of the input query. For indexing algorithms, in addition to using proximity phrases, there is also a step to assess the semantic relevance of keywords. This makes the indexing more valuable, especially with the queries which do not contain verbatim texts in the document.

3 Result Evaluation

In this article, the coreference method of handling the Vietnamese query and the semantic model of text of the authors is an additional support method for search engines, helping search enough semantic content that it might have, instead of just searching for the keywords of the query. To evaluate the effectiveness of the proposed model, the authors based on the average "loss function" parameter, signed as L, calculated by the following formula:

$$L = \frac{1}{T}\sum_{t=1}^{T} |y' - y^t| \ (5)$$

In which T is the number of test samples, y^t is the actual labeled value of the sample, y' is the ranking value predicted by the model.

The authors compare the proposed model with the VSM Algorithm. By surveying the queries, the evaluation results are given by the table of L value as follows:

Table 2. Test Results of the System

No.	Query			Number of test samples	Loss function	L
	O	V	O		VSM Algorithm	Proposed model
1	MU	borrowed	Falcao	120	1.8333	1.67
2	Arsenal, Man City	fight	Super Cup in England	150	4.6667	1.14
3	MU	draw	Sunderland	100	3.5000	0.50
4	Lampard	score		200	2.5714	1.00
5	Mu	borrowed	Falcao	150	4.1667	1.17
	Average L				**3.3476**	**1.0952**

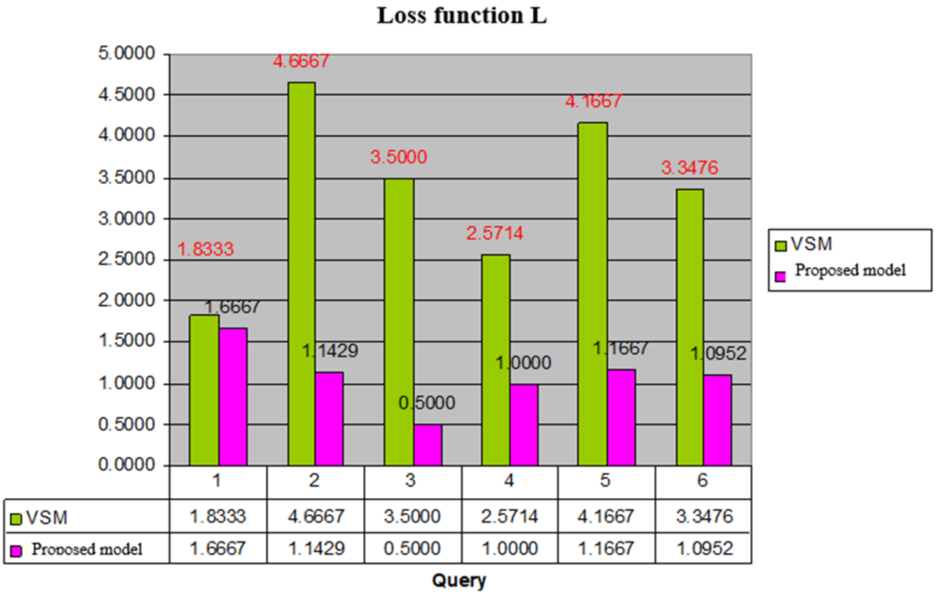


Figure 5. Evaluation Results between the Two Models

It can be seen that the results from semantic model show the indexing results with much lower semantic deviation than the indexing results by VSM model.

4 Conclusion

Some words may not have a semantic relation but can still be coreferent owing to their semantic similarity. This observation has led Ponzetto and Strube (11) to encode features based on various measures of similarity, which have been shown to improve their baseline system.

While using semantic roles improves Ponzetto and Strube's resolver (11), semantic parallelism is a fairly weak indicator of coreference. For instance, if two verbs denote events that are unrelated to each other, it is not clear why their arguments should be coreferent even if they have the same semantic role.

Generally speaking, the results of employing semantic and world knowledge to improve knowledge-poor coreference resolvers are mixed. The mixed results can be attributed at least in part to differences in the strengths of the baseline resolvers employed in the evaluation: the stronger the baseline is, the harder it would be to improve its performance. Since different researchers employed different baselines and evaluated their resolvers on different feature sets, it is not easy to draw general conclusions on the usefulness of different kinds of semantic features. We presented an overview of the models and features developed for learning-based entity coreference resolution. Despite the continued progress on this task, it is far from being solved. Recent results suggest that the performance of coreference models that do not employ sophisticated knowledge is plateauing. (12) Hence, one of the fruitful avenues of future research will likely come from the incorporation of sophisticated knowledge sources.

As coreference resolution is inherently a clustering task, it has received a lot of attention in the machine learning and data mining communities, where the task has been tackled under different names, such as record linkage/matching and duplicate detection. Some researchers have focused on name matching, where the goal is to determine whether the names appearing in two records in a database refer to the same entity. The focus on name matching effectively ignores pronoun resolution and common noun phrase resolution, which are arguably the most difficult subtasks of entity coreference resolution. (13)

Many machine-learned entity-based models have been developed over the years. The most notable ones include the entity-based versions of mention-pair models and entionranking models. Entity-mention models, the entity-based version of mention-pair models, determine whether a mention is coreferent with a preceding, possibly partially-formed, cluster. (14, 15) Despite their improved expressiveness, early entitymention models have not yielded particularly encouraging results. Cluster-ranking models, on the other hand, are the entity-based version of mention-ranking models. (16) They rank preceding clusters rather than candidate antecedents, and have been shown to outperform entity-mention models, mention-pair models, and mention-ranking models. While the entity-based models discussed so far have all attempted to process the mentions in a test text in a left-to-right manner, easy-first models aim to make easy linking decisions first, and then use the information extracted from the clusters established thus far to help identify the difficult links.

Literature:

1. Ngo VM, Cao TH. A Generalized Vector Space Model for Ontology-Based Information Retrieval. Vietnamese Journal on Information Technologies and Communications. 2009; 22(2):43-53.
2. Lee H, Chang A, Jurafsky D, Peirsman Y, Chambers N, Surdeanu M. Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. Stanford University, University of Leuven, United States Naval Academy; 2012. 32 p.
3. Denis P, Baldridge J. A ranking approach to pronoun resolution. Proceedings of the 20th International Joint Conference on Artifical intelligence (IJCAI 2007). 588-1593 p.
4. Hoste V. Manual for the Annotation of Coreferences in Dutch Newspaper Texts. 2005. 258 p.
5. Mccarthy JF. A trainable approach to coreference resolution for information extraction. 1996. 198 p.
6. Chia Hung Lin, Chia-Wei Yen, Jen-Shin Hong, Cru-Lara S. Event-Base textual document retrieval by using semantic role labeling and coreference resolution. 2007. 7 p.
7. Andreev AM, Berezkin DV, Syuzev VV, Shabanov VI. Models and methods of automatic classification of text documents. Vestn. MSTU. Ser. Instrument making. 2003; 3:98–108.
8. Andreev AM. Automatic text classification using neural-nets algorithms and semantic analysis. In Berezkin DV, Morozov VV, Simakov KV (Eds.). Proceedings of the fifth All-Russian scientific conference (RCDL'2003). St. Petersburg: Research Institute of Chemistry, St. Petersburg State University; 2003. 140–149 p.
9. Schenkel R, Broschart A, Hwang S, Theobald M, Weikum G. Efficient Text Proximity Search - SPIRE'07 Proceedings of the 14th international conference on String processing and information retrieval. 2007. 287-299 p.
10. Rocha C, Schwabe D, de Aragao MP. A hybrid approach for searching in the semanticweb. Proceedings of the 13th international conference on World Wide Web. 2004. 374–383 p.
11. Ponzetto SP, Strube M. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. Proceedings of the Human Language Technology Conference and Conference of the North American Chapter of the Association for Computational Linguistics. 2006. 192–199 p.
12. Wiseman S, Rush AM, Shieber SM. Learning global features for coreference resolution. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016. 994–1004 p.
13. Ng V, Cardie C. Improving machine learning approaches to coreference resolution. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2012. 104–111 p.
14. Luo X, Ittycheriah A, Jing H, Kambhatla N, Roukos S. A mention-synchronous coreference resolution algorithm based on the Bell tree. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. 2004. 135–142 p.
15. Yang X, Su J, Zhou G, Tan CL. An NP-cluster based approach to coreference resolution. Proceedings of the 20th International Conference on Computational Linguistics. 2004.
16. Rahman A, Ng V. Supervised models for coreference resolution. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. 2009. 968–977 p.

Primary Paper Section: I

Secondary Paper Section: IN, JC, JD