Received: January 15, 2018

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

WERTES: Web as External Resources for Textual Entailment Systems

Abdiansah Abdiansah[1*] **Azhari Azhari[2]** **Anny Kartika Sari[2]**

[1]*Universitas Sriwijaya, South-Sumatera, Indonesia*
[2]*Universitas Gadjah Mada, Yogyakarta, Indonesia*
* Corresponding author's Email: abdiansah@unsri.ac.id

Abstract: Research in Textual Entailment (TE) has been widely conducted, mainly in natural language based systems, since TE can provide solutions to semantic problems. Usually, the researchers focus on method improvement, hence, they use standard data sets, which are specific to a particular language, primarily in English. For low-resource languages, it is very difficult to find data sets to test the TE systems. Therefore, in this paper we propose a model to extract data from the web to serve as data set for TE systems. The model can be used for cross-language domains with simple modifications. Two datasets are created and used to evaluate the model, i.e. DS-100-R, which contains facts, and DS-100-W, which contains non-facts. The model produces a set of sentences that are expected to be relevant to the queries. Some algorithms are created to address problems that arise during experiments. Based on the evaluation, the model accuracy for DS-100-R dataset is 79.0%, and for DS-100-W dataset is 70.0%. Hence, the overall model accuracy is 74.5%.

Keywords: Textual entailment, Low-resources, Web.

1. Introduction

Research in the field of Textual Entailment (TE) was pioneered by Dagan and Glickman in 2004 [1], and is still actively conducted by some scientists [2]. In general, the problem faced in this field is how to recognize that the meaning of a text can be expressed or inferred by another text. This issue is similar to the common problem that exists in Natural Language Processing (NLP), which is the variability, where the same meaning can be formed or composed of different sentence structures. With these relations, the solutions obtained in TE can also be applied to NLP-based systems such as Question Answering System (QAS), Machine Translation (MT), Information Extraction (IE) and others.

Evaluation using standard data set is required to track the development of TE methods [3]. Unfortunately, the standard dataset is available only for particular languages, especially in English. For low-resource languages such as Indonesian, standard dataset is not available. Moreover, the use of different target language may influence the NLP strategies and techniques to be employed. Therefore, TE evaluation that focuses on method improvement is well suited using standard dataset, but if the method is applied to low resource languages, it will be difficult to find standard dataset.

The Web is a large and growing source of data, but the available data is unstructured or semi-structured. A lot of research has been conducted to explore and retrieve data from the Web to be utilized as a secondary source of knowledge [4]. However, until now there has been no agreement among the researchers to develop an independent framework and offer it as standard. The techniques to be used are still for specific needs and in particular domains [5]. The biggest challenge is how to convert unstructured data into structured data so that the data can be utilized as dataset for a system. One of the advantages of the use of the Web as an external resource is that the availability of large amounts of data is guaranteed. Data collection task will not be a problem for researchers, hence, they can focus on the next stage.

Languages with low-resources can be barriers in TE research. Therefore, some scientists conduct research to produce a dataset containing T and H pairs automatically using their native languages such as Spanish [6], Arabic [7, 8], German [9] and Czech [10]. But for the Indonesian language. But for the Indonesian language, to our knowledge there is not yet explored. To fill the gap, we propose a model called WERTES (Web as External Resources for Textual Entailment Systems) to retrieve data of the required language from the Web and convert it into TE dataset. The model will take H as input to generate as many T-H pairs as possible. The Web serves as a data source to find T. Each T-H pair is expected to give a positive value. Although the system is built specifically for Indonesian language, modifications for other languages are not complicated, hence, it can be used for cross-languages. The system is evaluated using another system that is called TES (Textual Entailment System). The evaluation procedure refers to the AVE (Answer Validation Exercise) field [11–13], which is by labelling VALID for positive T-H pairs, and labelling REJECTED for negative pairs. The overall results are measured by accuracy.

This article is structured as follows. Section 2 contains a brief summary of research in TE and recent studies related to the dataset and the used methods. Section 3 presents the proposed model and the explanation of each component. Section 4 explains the dataset and evaluation specifications utilized in the experiments. Section 5 discusses the experimental results, and the last section contains conclusions.

2. Related work

Many TE systems are developed using a standard dataset with English as the target language. However, there are also TE systems developed in other languages. For example, Spanish [6], Arabic [7, 8], German [9], and Czech [10], Italian [14], Japanese [15], China [16]. Moreover, some researchers build systems are independent from standard dataset, although the experimental data still refers to the standard dataset [17, 18]. These all works indicate that the research in TE field still grows [2]. In the topic of generated TE corpus, different language has different methodology. Based on our literature study, we found some similar works have been done by scientists [6, 8, 19].. Next paragraph we will explain brief of summary of their research.

Burger & Ferro [19], generate a large corpus of TE pairs (100.000 pairs) from the lead of paragraph and headline of English news articles. They manually inspected a small of set of news stories in order to locate the most productive source of entailments, then built an annotation interface for rapid manual evaluation of further exemplars. They manually inspected over 200 news stories from 11 different sources and observed the headline of a news article have entailment relationship with lead paragraph. For each headline or lead paragraph pair, a human rendered a judgment of YES (entailment), MAYBE (close to/not be entailment) and NO (no entailment). Their experiment results show that the MiTAP corpus (111 pairs) is YES (54 pairs/49%), NO (39 pairs/35%), and MAYBE (18 pairs/16%). Whereas in the Gigaword corpus (103 pairs) is YES (52 pairs/50%), NO (37 pairs/36%), and MAYBE (14 pairs/14%).

Penas et al. [6] development of SPARTE, a corpus for training and testing RTE systems in Spanish, and specially, systems aimed at validating the correctness of the answers given by QAS (Question Answering System). SPARTE has been built from the Spanish corpora used at Cross-Language Evaluation Forum (CLEF) for evaluating QAS during 2003–2005. The first step to build SPARTE is to turn the questions into an affirmative form and assessment the candidates answer by a human: correct (R), incorrect (W), inexact (X) or unsupported (U). Once the answers are grouped as possible instances for building the hypothesis, the next step is to build the text-hypothesis pairs with the entailment TRUE/FALSE value. The final SPARTE corpus has 2.962 text-hypothesis pairs from 635 different questions with the number of pairs TRUE is 695 and the number of pairs FALSE is 2.267. They performed a partial human evaluation of the corpus in order to assess the quality of SPARTE. They took randomly the 10% (70 pairs) of TRUE pairs and the 5% (113 pairs) of the FALSE ones. The results of pairs TRUE is 67 pairs (96%) is correct and 3 pairs (4%) is incorrect, whereas 111 pairs (98%) is correct and 2 pairs (2%) is incorrect for the pairs FALSE.

Alabbas [8] follows [19] with different techniques, they developed a semi-automatic technique for creating a first dataset for TE systems for Arabic using an extension of the ‘headline-lead paragraph’ technique. The technique consists of two tools, the first tool is responsible for automatically collecting T-H pairs from news websites, whereas the second tool is an online annotation system that allows annotators to annotate their collected pairs manually. T-H pairs automatically acquired from

Arabic newspapers' and TV channels' websites as queries to be input to Google via the standard Google-API. Then, they select the first paragraph, which usually represents the most related sentence(s) in the article with the headline. This technique produces a large number of T-H pairs without any bias in either T's or H's. Next, all pairs are annotated performed by human using 'YES' (entailment), 'NO' (no entailment), and 'UN' (unknown) labels. The final dataset, namely Arabic TE dataset (ArbTEDS), consists of 618 T-H pairs. They used two evaluation based on the number of annotators agree (see Table 1 in [8]). In '2 agree' (an annotator agrees with at least one co-annotator) the results are YES (478 pairs/80%) and NO (122 pairs/20%). Whereas '3 agree' the results are YES (409 pairs/68%) and NO (69 pairs/12%). The rest of results is UN.

3. Proposed model

We present our proposed model as WERTES, which architecture is shown by Fig. 1. The input of the system is question-answer pairs because the test-bed is conducted in the QAS area with focus on answer validation using textual entailment. The target language is a low resource language. In our case, the target language is Indonesian. This model can also be used for other languages by adjusting the algorithms related to language processing. The question-answer pairs will be processed by AHG (Automatic Hypothesis Generation) to generate H. QG (Query Generation) generate a set of queries which are then submitted to the Search Engines to search for relevant data from the Web. The results are in the form of HTML files, which are then extracted by the Sentence Extraction (SE) to generate a set of sentences. The output of WERTES is a set of sentences (T) that are considered to have entailments with queries (H). The output will be evaluated by a Textual Entailment System (TES) that determines whether each of the T-H pairs is VALID (T entails H) or REJECTED (T does not entail H).

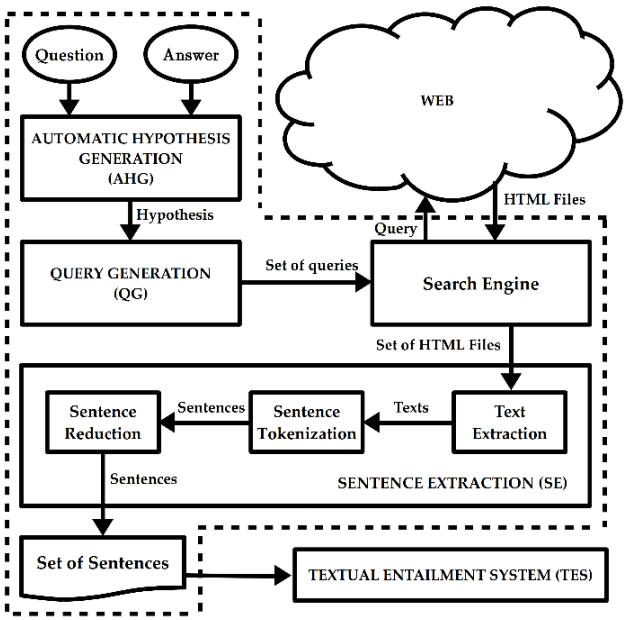

Figure. 1 Architecture of WERTES

3.1 Query generation

The Query Generation component is responsible to generate new queries from the original one. Query modification is conducted by partly subtracting the original query information, and the results are new queries. Technically, this is done by deleting word by word of the original query, starting from the rightmost word. We name this technique as Right-first Cutting (RfC). Details of the RfC algorithm can be seen in ALGORITHM 1. The output of the algorithm is a set of queries that will be sent to search engine.

ALGORITHM 1: RfC Algorithm

```
Q ← query
Q_temp ← Q
Q_result ← add Q
while the number of words in Q_temp > 2 do
    temp ← delete a right word from Q_temp
    Q_result ← add temp
    Q_temp ← temp
end while
return Q_result
```

Here is an example of query generation from original query. Consider the following original query:

- Q: "Soekarno presiden pertama Indonesia" (Soekarno is the first president of Indonesia)

After Q is processed by QG, the generated queries are as follows:

- Q1: "Soekarno presiden pertama Indonesia" (Soekarno is the first president of Indonesia)
- Q2: "Soekarno presiden pertama" (Soekarno is the first president)
- Q3: "Soekarno presiden" (Soekarno is the president)

The total number of queries generated by the algorithm is n(Q) – 1, where n(Q) is the number of the words in the original query. In the previous example, n(Q) = 4, so, the total number of the generated queries is 3. All of the generated queries will be sent to the search engine. This component is evaluated by comparing with other approaches, namely the baseline (without cutting) and cutting word from left or Left-first Cutting (LfC). In Section 5.1 we discuss the results of experiments in details. An analysis is also conducted to see if the results of the generated queries are relevant to the original query.

3.2 Sentence tokenization

Sentence tokenization serves to produce sentences from the corpus. A simple technique commonly used is point detection, as usually a sentence ends with a point. However, the technique cannot distinguish a point that ends a sentence and a point with specific purpose such as academic titles (Prof., Dr.) and abbreviations (Jan., Aug.). Therefore, our model uses Punkt algorithm [20] to detect sentence boundaries.

ALGORITHM 2: Sentence-expansion Algorithm

```
Q ← a query
S ← set of sentences in array of string
Th ← threshold (th = 3)
Buff ← empty array of string
NewSen ← empty array of string
Counter ← 0
for the first sentence of S to the last sentences do
    Buff ← add S[i]
    if Buff contains all the words of Q then
        NewSen ← add Buff
        Buff ← empty
        Counter ← 0
    end if
    Counter ← Counter + 1
    if Counter = Th then
        Buff ← empty
        Counter ← 0
    end if
end for
return NewSen
```

Tokenization with Punkt algorithm gives better results than point detection technique, however, there are special cases that must be considered. For example, there are a query and a document to be tokenized. We need to determine the relevancy between the document and the query by examining whether all words forming the query are included in at least one of the sentences produced from the document. The result shows that none of the sentences contains the all words, even though the document contains all words in the query. To solve the problem, we modified the tokenization method. The basic idea is "a sentence may have a related meaning with the neighbor sentences. To obtain the full meaning, each sentence pair are combined into one". Based on the idea, we develop an algorithm which is called Sentence-expansion algorithm, which is presented in ALGORITHM 2.

Sentence-expansion algorithm receives the input of a query and set of tokenization results. Sentence retrieval is done sequentially from the first sentence to the last sentence. Each sentence will be stored into Buff, which serves as a temporary storage. There are two conditional questions to be used to store the result of sentence: (1) If Buff contains all words in the query, then save Buff into NewSen which stores relevant sentences, then empty the content of Buff; and (2) If the number of sentences in Buff equal to the threshold value, then empty the content of Buff. The threshold value is 3, which is derived from the assessment to the results of text extraction in this research.

3.3 Sentence reduction

Sentence Tokenization component gives a set of relevant sentences with their respective queries. In most cases, long sentences are produced, which makes them similar to a paragraph. This increases computation time, especially for calculating relevancy between queries and sentences. Long sentences also potentially decrease the relevancy because too many words are involved in the calculation. To solve these problems, we develop the Sentence-reduction algorithm, as presented in ALGORITHM 3. The basic idea of the algorithm is "The words in sentences that are irrelevant to the words in the query will be discarded using systematic technique". For each sentence, the algorithm will make two delimiters, i.e. left and right borders. The words located in between the left and right borders will be considered as relevant sentence, while the rest will be discarded. Sentence-reduction algorithm used word delimiter as the benchmark when discarding irrelevant words without considering another factor. It potentially may cause the original sentence to lose its meaning. Therefore, an experiment is conducted to see the percentage of sentences whose meaning do not alter

after sentence reduction process. Details of the experiment is presented in Section 5.4.

ALGORITHM 3: Sentence-reduction Algorithm

```
Q ← a query
S ← a sentence
Left_border ← 0
Right_border ← 0
Buff_Q ← empty array of string
for the first word of S to the last word of S do
    word ← S[i]
    if word in Q and Left_border = 0 then
        Left_boder ← i
        Buff_Q ← add word
    else if word in Q and word not in Buff_Q then
        Buff_Q ← add word
        if length(Buff_Q) = length(Q) then
            Right_border ← i
            break
        end if
    end if
    if i = length(S) then
        Right_border = i
    end if
end for
return subset (S, Left_border, Right_border)
```

4. Dataset and evaluation specification

QAS which focus on answer validation will be used as test-bed system for the result of WERTES. The dataset is used contains a set of question-answer pairs annotated directly by humans and divided into DS-100-R and DS-100-W. DS-100-R contains 100 question-answer pairs with correct answers, while DS-100-W contains 100 question-answer pairs with incorrect answers. The overall result is obtained from the average value of the two data sets. The format of data set is {*question*, *answer*}. An example of an item in the data set is {*who is the first president of indonesia*, *soekarno*}.

TES uses Answer-validation algorithm presented by ALGORITHM 5 to validate the hypothesis. There are two main processes in the algorithm: (1) *Labelling*, where T will be labelled [RELEVANT] if H is a subset of T; and (2) *Validation*, where there are two determinant variables: α which contains the number of T relevant to H, and β which contains the actual value. The value of β will be TRUE if a hypothesis is factually true. Otherwise, it is FALSE. The value of β is determined based on the data set type. If H is from DS-100-R then β is TRUE, if it is from DS-100-W then β is FALSE. H is VALID if one of the following two conditions is met: (1) at least one element of T is a superset of H, and the value of H is true (β = True); or (2) no element of T is a superset of H, and the value of H is false (β = False). If none of the conditions is met, H is REJECTED. Based on the decision by WERTES, the decision accuracy for each dataset is measured using the following equation:

$$accurracy = \frac{Total\ number\ of\ VALID\ hypothesis}{Total\ number\ of\ hypothesis} \quad (1)$$

ALGORITHM 5: Answer-validation Algorithm

```
H ← a hypothesis
T ← set of texts
for the first text of T to the last text of T do
    if H Subset T then
        T[i] = add label [RELEVANCE]
    end if
end for
alpha ← Numbers of T that relevant with H
beta ← Truth value of hypothesis
if (alpha>0 & beta=True) || (alpha=0 & beta=False) then
    return H is VALID
else
    return H is REJECTED
end
```

5. Experiment and results

This section presents the experiment results for each component as well as for the whole system. There are four components to be evaluated: (1) *Query Generation*; (2) *Search Engine*; (3) *Sentence Tokenization*; and (4) *Sentence Reduction*. The following sub sections describe the experiments results in details.

5.1 Evaluation on query generation

The QG algorithm uses the query cutting technique from the right side because it produces more sentence than query cutting technique from the left side. We proved this in our first experiment. In the experiment, two pairs of queries have been employed. The queries used in the experiment were as follows:

- Q_1: “soekarno presiden pertama indonesia”
- Q_2: “megawati presiden pertama”
- Q_3: “candi borobudur terletak magelang”
- Q_4: “candi borobudur terletak yogyakarta”

Table 1. The number of sentences produced for query generation using different methods

Queries	Number of sentences		
	Baseline	LfC	RfC
Q_1	12	62	164
Q_2	8	54	69
Q_3	20	71	306
Q_4	4	22	291

Table 2. Results of query modification experiment using specific keyword

Queries	Number of raw sentences	Distributed Number of sentences	File size (MB)
Q(baseline)	62	26 (13%)*	639
QWi	49	22 (11%)	603
QWo	87*	26 (13%)*	546*

The queries have been processed using three techniques: (1) *non-cutting*, serving as the baseline experiment; (2) *LfC (Left-first Cutting)*, in which the query sentences are cut from the left side of the query; and (3) *RfC (Right-first Cutting)*, in which the query sentences are cut from the right side of the query. WERTES then processed the queries to obtain a set of sentences taken from the web. Table 1 shows the experiment results, in which RfC produced the most number of sentences compared to other methods.

The baseline produced the lowest number of sentences because it only uses single query. Compared to the baseline, LfC increases the number of sentences to 375%, while RfC increases the number of sentences to 1,786%.

5.2 Evaluation on search engine

The experiment used 200 queries, which were taken from datasets DS-100-R and DS-100-W. In this experiment, Query Generation component was not used. There were three types of query to be tested: (1) Query/Q (baseline), i.e. using queries without adding keywords, which also serves as experiment baseline; (2) Query+Wikipedia/QWi, i.e. adding keyword "Wikipedia" to the query; and (3) Query+Wordpress/QWo, i.e. adding the keyword "WordPress" to the query.

Three aspects were examined from this experiment: (1) The total number of raw sentences obtained by each type of query; (2) Total number of clean sentences, which are sentences that contains query keyword(s) and obtained after system processing; and (3) The file size of the document containing the raw sentences.

Table 2 shows the experimental results of query modification. The best value is indicated by an asterisk symbol (*). The higher the values of both the number of raw and clean sentences, the better is the result. Conversely, the smaller the file size, the better is the result as the computation time can be reduced. Since QG component was not used in this experiment, the number of clean sentences to be produced from each type of query was very small.

The experimental results show that query in the form of 'Query+Wordpress' outperformed others. In terms of the number of the clean sentences to be produced, its performance was the same as the baseline. However, in terms of the number of the raw sentences to be produced and the file size, it was better than the other two types of queries. In spite of the small file size, the 'Query+Wordpress' query was able to produce rawest sentences as well as clean sentences. Query in the form of 'Query+Wikipedia' yielded the worst result. This is due to the different HTML structure Wikipedia compared to blogging system. The experiment proves that text extraction using tag <p> for blogging systems is suitable to be used in our model.

5.3 Evaluation on sentence tokenization

The experiment used dataset DS-100-R and DS-100-W that contained 200 queries. The baseline was sentence tokenization using standard tokenization techniques (Punkt). The experiment specifications followed the specifications described in Section 4. We evaluated the sentences produced by WERTES. The sentences were evaluated by TES to calculate the number of relevant sentences as well as the accuracy values. The experiment results can be seen in Table 3.

Table 3 shows that the results of tokenization with sentence expansion outperformed the baseline in all aspects. The number of sentences increased from 9,914 to 10,350 (about 4.4%), while the number of relevant sentences increased from 175 to 675, which is about 285.7%. The accuracy value increased about 8.5%, which proves that the sentence expansion approach is able to improve the accuracy of the system significantly.

Table 3. Results of sentence tokenization

Techniques	Total number of sentences	Number of relevant sentences	Accuracy
Punkt (baseline)	9.914	175	66,0%
Punkt+Sentence-expansion	10.350	675	74,5%

Table 4. Results of sentence reduction

Types of sentences	Number of words	Number of relevant sentences
Original sentences	7,487	88
Reduction sentences	3,737	99

Table 5. Result of WERTES evaluation on datasets

Datasets	WERTES's output	TES's output	
	Number of sentences	Number of relevant sentences	Accuracy
DS-100-R	5.409	586	79,0%
DS-100-W	4.942	89	70,5%

5.4 Evaluation on sentence reduction

The experiment used top five queries from each of the DS-100-R and DS-100-W data sets, hence, there were 10 queries to be used. The queries were processed by WERTES to produce a set of sentences which were then evaluated by TES. We calculated the total number of words and the number of relevant sentences produced by either the original sentence and the reduced sentence.

Table 4 shows the result of the experiment. It can be concluded that sentence-reduction algorithm has removed 3,752 words, which is around 50.0% of the original words. From manual comparison, 11 sentences were irrelevant. Hence, the percentage of relevant sentences produced by the reduced sentences is 88.9%. Ambiguity is one of the problems in the reduced sentences that caused a failure to find relevant sentences. For example, the word "presiden" and "presiden-nya" are considered the same by the system, whereas the meaning is actually different. The other problem is that the system cannot distinguish the relationship between words. For example, the word "first" in the phrase "first president" and "first organization" is assumed to be the same. Nevertheless, the percentage of irrelevant sentences is not significant compared to the benefit of sentence reduction. Therefore, we consider that the Sentences-reduction algorithm should be used in WERTES.

5.5 WERTES Evaluation

The previous sub-sections describe the experimental results of sub-components. The evaluation result of each sub-component is as satisfying as expected. To evaluated the whole system, we conducted another experiment, which is explained in this sub-section. The evaluation used datasets DS-100-R and DS-100-W, which means that the total number of queries processed by the system is 200 queries. The experimental specifications followed the specifications described in Section 4. The results of the experiment can be seen in Table 5.

From Table 5, it can be seen that the number of relevant sentences produced from DS-100-R is 586. This means that only about 10.88% of sentences are considered relevant by TES. Data set DS-100-W gives even a smaller percentage, which is about 0.18%. This occurs because the sentences produced by WERTES are based on local relevancy, where they are relevant only to the given query (see Section 3.4.2). The locally relevant sentences resulted from the original query will surely be detected relevant by TES, but not necessarily for new queries (see Section 5.1). In the table, it also can be seen that the number of relevant sentences produced from DS-100-R data set is more than from DS-100-W data set. It means that in the domain of Indonesian history, most information found on the Internet is correct because it is based on facts. Therefore, the information that is available on the Internet can serve as the knowledge support for knowledge-based systems.

To determine whether a sentence is relevant or not, ALGORITHM 5 is used. In short, if a query comes from DS-100-R and yields relevant sentences then it will be considered as valid, otherwise it will be rejected. If a query comes from DS-100-W and it yields no relevant sentences, then the query is considered valid, otherwise it will be rejected. The accuracy of DS-100-R is 79.0%, which indicates that 79 queries are valid out of 100 queries, while the rests are rejected. The overall system accuracy is 74.5%, which is obtained from the average accuracy of DS-100-R and DS-100-W. The experiment result is as what we expected. DS-100-R should produce high number of relevant sentences because each query (hypothesis) is formulated from the original question asked by the user and the correct answer for the question. On the other hand, DS-100-W is expected to produce as few relevant sentences as possible because each query is formulated from the original question asked by the user and an incorrect answer for the question. This result shows that WERTES is able to process information available on the Web as supporting facts for textual entailment systems.

It is very difficult to direct comparison between our result and the others because each approaches have different methodology and strategy. Although the basic concept is same, but the data sources,

Table 6. The comparison results of pairs

Approaches	Datasets	Corrects	Incorrect
Penas et al. [6]	Pairs TRUE	96%	4%
WERTES	DS-100-R	79%	21%
Penas et al. [6]	Pairs FALSE	98%	2%
WERTES	DS-100-W	71%	29%

techniques, and evaluation are very different. Unfortunately, we can not compare our results to [8, 19] because they are not measure the accuracy, they just anotated the data. Nevertheless, our result can be compared to [6] because they measure how many pairs are correct and incorrect (accuracy). In Table 6 can be seen that our result is inferior than [6] but the result is relative to the dataset. Our accuracy is above 70% and it can be considered quite satisfactory to be made as research baseline. Moreover, our work can be stepping-stone for the further research of Indonesian TE.

6. Conclusion

A model to extract data from the web to serve as data set for TE systems has been built. The model retrieves a set of texts T from the web using hypothesis H as the input. WERTES is the system that has been built based on the model. The system consists of 4 main components, namely Automatic Hypothesis Generation, Query Generation, Search Engine and Sentence Extraction. The target language is Indonesian because the language is one of low resource languages. However, the model can be applied to other low resource languages as well with some adjustments.

To evaluate the model, we conducted experiments to components or sub-components, as well as the whole system. Experiments to components or sub-components were aimed to prove the claims and check the feasibility of the proposed methods, while experiment to the whole system was to test if WERTES gave result as expected. The system accuracy is 74.5%, which is obtained from the average accuracy of DS-100-R and DS-100-W data sets. This shows that WERTES can be used to extract information from the Web to provide facts for textual entailment systems. For future work, we will address the facts provided by WERTES to validate answers in question answering systems. Furthermore, the application of the model to other low resource languages may be interesting to examine.

References

[1] I. Dagan and O. Glickman, "Probabilistic textual entailment: Generic applied modelling of language variability", *Learn. Methods Text Underst. Min*, pp. 26-29, 2004.

[2] L. Bentivogli, I. Dagan, and M. Bernardo, "The Recognizing Textual Entailment Challenges: Datasets and Methodologies", In: *Handbook of Linguistic Annotation*, pp. 3-4, 2017.

[3] I. Androutsopoulos and P. Malakasiotis. "A Survey of Paraphrasing and Textual Entailment Methods", *J. Artif. Intell. Res*, pp. 135–187, 2010.

[4] A. Gangemi, D. R. Recupero, M. Mongiovì, A. G. Nuzzolese, and V. Presutti, "Identifying motifs for evaluating open knowledge extraction on the Web", *Knowledge-Based Syst*, pp. 33–41, 2016.

[5] E. Ferrara, P. D. Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey", *Knowledge-Based Syst*, pp. 301–323, 2014.

[6] A. Peñas, A. Rodrigo, and F. Verdejo, "Sparte, a test suite for recognising textual entailment in Spanish", In: *Proc. of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 275-286, 2006.

[7] K. Garoufi, "Towards a Better Understanding of Applied Textual Entailment", *Master Thesis*, Saarland University, Saarbrücken, Germany, 2007.

[8] M. Alabbas, "A Dataset for Arabic Textual Entailment", In: *Proc. of the Student Research Workshop associated with RANLP 2013*, pp. 7-13, 2013.

[9] B. D. Zeller and S, Padó, "A search task dataset for German textual entailment". In: *Proc. of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pp. 288-299, 2013.

[10] Z. Nevěřilová, "Paraphrase and textual entailment generation in Czech", *Computación y Sistemas*, Vol. 18, No. 3, pp. 555-568, 2014.

[11] A. Peñas, Á. Rodrigo, V. Sama, and F. Verdejo, "Overview of the Answer Validation Exercise 2006", *Lecture Notes in Computer Science*, pp. 257–264, 2007.

[12] A. Peñas, Á. Rodrigo, and F. Verdejo, "Overview of the Answer Validation Exercise 2007", *Advances in Multilingual and Multimodal Information Retrieval*, pp. 237–248, 2008.

[13] Á. Rodrigo, A. Peñas, and F. Verdejo, "Overview of the Answer Validation Exercise 2008", *Lecture Notes in Computer Science*, pp. 296–313, 2009.

[14] L. Bentivogli and B. Magnini, "An Italian Dataset of Textual Entailment Graphs for Text

Exploration of Customer Interactions", In: *Proc. of the First Italian Conference on Computational Linguistics*, pp. 63–66, 2014.
[15] S. Matsuyoshi, Y. Miyao, T. Shibata, C. Lin, C. Shih, Y. Watanabe, and T. Mitamura, "Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task", In: *Proc. of the 11th NTCIR Conference*, pp. 223–232, 2014.
[16] M. Day, C. Tu, S. Huang, H. Vong, and S. Wu, "IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-9 RITE", In: *Proc. of the 9th NTCIR Workshop Meeting*, pp. 462–468, 2013.
[17] M. Víta and V. Kriz. "Word2vec Based System for Recognizing Partial Textual Entailment", In: *Proc. of the Federated Conference on Computer Science and Information Systems*, pp. 513–516, 2016.
[18] R. Zanoli and S. Colombo, "A transformation-driven approach for recognizing textual entailment", *Nat. Lang. Eng. FirstView*, pp.1–28, 2016.
[19] J. Burger and L. Ferro, "Generating an entailment corpus from news headlines", In: *Proc. of the ACL workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pp. 49-54, 2005.
[20] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection", *Comput. Linguist*, pp. 485–525, 2006.