NEPAL: A Tool for Never-Ending Paraphrase Learning

Paulo César Polastri[1], Helena de Medeiros Caseli[1], and Eloize Rossi Marques Seno[2]

[1] Federal University of São Carlos (UFSCar), Brazil,
pcpolastri@hotmail.com, helenacaseli@dc.ufscar.br
[2] Federal Institute of São Paulo (IFSP), Brazil,
eloizeseno@gmail.com

Abstract. Use different words to express/convey the same message is an ordinary task in any language and one of the best ways to do so is using paraphrases. As a consequence, a proper treatment of paraphrases is crucial for several NLP applications, such as Machine Translation, Multidocument Summarization and Natural Language Generation. This paper describes NEPAL, an automatic system capable of learning paraphrases in Brazilian Portuguese. The extraction is made from a bilingual parallel corpus composed of news that are available online. To do so, NEPAL applies the [5]'s never-ending machine learning approach together with the paraphrase extraction method proposed by [2]. The experiment described in this paper show promising results achieving 86% of correctly extracted paraphrases.[3]

1 INTRODUCTION

Paraphrases can be defined as alternative ways to convey the same information using different linguistic expressions [2, 4]. While humans can deal with these linguistic variations, computers, on the other hand, do not have the same ability as humans. Trying to solve this problem, many paraphrase recognition approaches have been proposed in the literature in recent years, mainly for English [14, 2, 4, 11].

The recognition and extraction of paraphrases are important tasks for several NLP applications. In Question Answering (QA), for example, a QA system that is able to handle paraphrases will also be able to properly answer variations of the same question [14]. In Multidocument Summarization systems, the recognition of paraphrases is useful for identifying redundant information in a document collection that needs to be summarized [3]. In Machine Translation systems, paraphrases can help the production of more accurate translations by, for example, increasing the text coverage replacing an unknown source word/n-gram by a paraphrase and performing the translation to the target language from this paraphrase [7]. In Automatic Sentence Fusion, the recognition of paraphrases can help in the production of more complete and objective sentences while it can also mitigate the redundancies [21].

According to [4], paraphrases are usually classified as: (i) lexical paraphrases, when formed by replacing words with equivalent words (see examples (1) and (3) in Table 1); or (ii) syntactic paraphrases, when there is a change in the syntactic structure (examples (2) and (5) in Table 1). Furthermore, paraphrases can occur in three levels of granularity:

[3] We would like to thank the grants #2013/11811-0 and #2013/50757-0, São Paulo Research Foundation (FAPESP).

(i) word level, when it encompasses only single words (examples (1) and (3) in Table 1); (ii) n-gram level, when it encompasses groups of words (example (4[4]) in Table 1); and (iii) sentence level, when it encompasses sentences (example (3) in Table 1).

Table 1. Examples of paraphrases

(1)	barrar (to bar)
	bloquear (to block)
(2)	A Honda construiu outro carro (Honda built another car)
	Outro carro foi construído pela Honda (Another car was built by Honda)
(3)	choro (cry)
	tristeza (sorrow)
(4)	Agulhas Negras (Agulhas Negras)
	Academina Militar Agulhas Negras (Agulhas Negras military Academy)
(5)	Letícia vendeu um sapato para Bia. (Letícia sold a shoe for Bia.)
	Bia comprou um sapato de Letícia (Bia bought a shoe from Letícia.)

This paper presents NEPAL[5] (Never-Ending Paraphrase Learner), a never-ending learning system developed to automatically learn lexical paraphrases from bilingual parallel corpus automatically crawled from the Web. This kind of corpus was chosen because it is the same used by [2], the method applied in our experiment. In [2], the idea is that an n-gram in a language A can be translated in different ways to language B (pivot language) and each of this translations in language B may be translated back to the language A. As a consequence, the correspondent n-grams in language A are considered to be paraphrases. An example is shown in Figure 1.

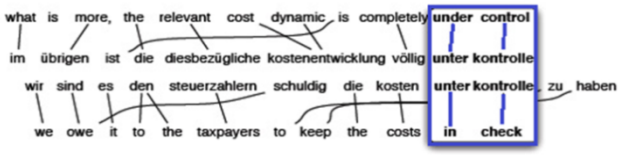

Fig. 1. Example of paraphrase extraction in English using German as pivot language [2]

This method was implemented in NEPAL together with the never-ending machine learning approach which aims at obtaining knowledge incrementally and steadily accumulating more knowledge over time. Following this approach, the accumulated knowledge is used to learn something new and it can even help to improve the system's learning ability. We choose this approach based on the good results shown by recent applications that use it as a means for extracting knowledge from large volume of information available mainly on the web [5].

So, NEPAL performs automatic recognition of lexical paraphrases in Brazilian Portuguese using English as pivot language following [2]. Based on the never-ending machine learning approach [5], the paraphrases recognized in previous iterations are used as a source of knowledge for learning new paraphrases. As a result, NEPAL produces

[4] "Agulhas Negras" is a proper name in Portuguese.

[5] Avaliable at: http://www.lalic.dc.ufscar.br/never-ending/nepal/.

a lexicon made of paraphrases in Brazilian Portuguese with an accuracy of 86% as verified in the experiment described in this paper.

2 RELATED WORK

For paraphrase extraction, NEPAL strongly follows the method of [2], in which paraphrases are extracted from bilingual parallel corpus based on well-known statistical machine translation (SMT) models [6]. As shown in Figure 1, the method assumes that an n-gram in a language A can be translated in various ways into language B (pivot language) and this translations in language B may be translated back to the language A, generating corresponding n-grams (paraphrases) in language A.

As it is possible to see in Figure 1, the n-gram "under control" in English was translated to German as "unter kontrolle" for which there is also another possible translation to English that is "in check". So, taken into account the alignments from source to target (source-target) and the other way back (target-source), the pair (under control, in check) is considered a paraphrase candidate.

Thus, the essence of this method is to align n-grams in a bilingual parallel corpus and cope with different n-grams in the source language that are aligned with the same n-gram in the pivot (target) language. Then, in a similar way to what happens in phrase-based statistical machine translation (PBSMT) approach, the method calculates the probability of each pair of alignments to be a paraphrase.

Among other studies conducted in order to recognize/extract paraphrases from a parallel corpus, in [11] is described the creation of a paraphrases database (PPDB), which comprises paraphrases extracted from the alignment of a parallel English-German bilingual corpus. To do so, the authors also applied the method proposed in [2].

As an extension of [2], [1] proposes a new method capable of recognizing paraphrases involving single words as well as n-grams containing up to seven words. This method uses what the authors called the "quasi-sense" annotation, which aims at solving some of the problems of the method [2] by restricting the meaning of the information.

Other methods were also proposed based on the use of parallel corpus but aiming at recognizing synonyms, such as [18]. In this work, the authors propose a method for recognizing synonyms, which they define as "single words that have the same idea".

3 NEVER-ENDING PARAPHRASE LEARNER

As already mentioned, NEPAL (Never-Ending Paraphrase Learner) is a never-ending machine learning system that aims at extracting paraphrases between single words from a bilingual parallel corpus. NEPAL was developed in Java and is divided into four modules which run in sequence and repeatedly in each iteration of NEPAL:

1. **Crawler** – This module is responsible for collecting texts from the web to build a parallel corpus. In our current prototype, in each iteration, around 40 different articles from the international version of the *Folha de São Paulo*[6] Brazilian newspaper are collected. This is a recent source of parallel texts for Brazilian Portuguese

[6] Available at: `www1.folha.uol.com.br/internacional/en/`.

with very high quality and in constant growing. The original news in Brazilian Portuguese are translated to English by human translators and the texts are feed constantly, an essential feature to enable the application of the never-ending approach.

2. **Dictionary builder** – This module is responsible for generating the bilingual dictionaries that will be used by the Extractor module. The dictionaries, in our version of [2]'s method, play the role of the alignment in source-target (pt-en) and target-source (en-pt) directions.[7]
   The Dictionary Builder has as input the pairs of parallel texts previously collected by the Crawler and generates as output two dictionaries: one pt-en (containing translations of Brazilian Portuguese words to English) and one en-pt (containing translations of English words to Brazilian Portuguese). Firstly, each pair of parallel texts is sentence aligned using the tools available at PorTAl[8]. After the sentence alignment of all pairs of parallel texts, Moses's [15] scripts are applied[9] and the lexical alignment is performed by GIZA++ aligner [17]. The entries containing stopwords in Brazilian Portuguese or English are removed from the dictionaries and the remaining words are stemmed and concatenated in unique entries.[10]
3. **Extractor** – The Extractor module is responsible for producing the pairs of paraphrases from the dictionaries. To do so, the method presented in [2] is followed. The output of this module is a list of quadruples (source_word1, pivot_word, source_word2, paraphrase_probability), where source_ word1 and source_word2 are aligned through pivot_word.
4. **Promoter** – The last module of NEPAL is responsible for promoting instances (paraphrase candidates) to beliefs (true candidates) using Weka [12] and a decision tree model J48 [19][11] The Promoter was trained with 1,800 instances manually annotated by two native speakers of Brazilian Portuguese giving rise to Promoter-0. The agreement between these two annotators was considered a good one with κ = 0.85 [8]. In the current version of NEPAL, only two features are used by the Promoter to "promote" an instance to belief (true paraphrase candidate):

   (a) **LCSR** (Longest Common Subsequence Ratio) calculated as the length of the longest common subsequence between two words divided by the length of the longest word [13]. LCSR values range from 0 to 1 and they are calculated between source_word1 and source_word2.

   (b) **Paraphrase probability** calculated as in [2]. The values here range from 0 to 100.

[7] In this paper, pt stands for Brazilian Portuguese and en stands for English.

[8] `http://www.lalic.dc.ufscar.br/portal/`.

[9] The Moses's scripts applied by NEPAL are: `tokenizer.perl`, `clean-corpus-n.perl`, `train-model.perl`

[10] The stemming was performed after the lexical alignment since according to [9], the lexical alignment between Brazilian Portuguese and English texts gives better results when surface forms (in spite of lemmas) are taken into account.

[11] The decision tree was the algorithm chosen here since it achieved the best precision compared to SVM [10] and Näive Bayes [16]. While the decision tree model achieved 89.74% accuracy in tests with 10-fold cross-validation, the SVM and the Näive Bayes achieved only 64.35% and 59.15%, respectively, in tests using the same dataset.

4 EXPERIMENT AND RESULTS

In our experiments, 15 iterations of NEPAL were run to see if there was any improvement in accuracy over time. In each iteration, 40 pairs of news were collected summing 600 pairs collected during the whole experiment. During the 15 iterations, three versions of the Promoter were generated. So, for this experiment, we evaluated: (1) the Promoter-0 trained as explained in section 3 (before the first iteration); (2) the Promoter-1, trained between the 5^{th} and 6^{th} iterations; and (3) the Promoter-2, trained between the 10^{th} and the 11^{th} iterations. It is important to say that the re-training of the model was made without any human supervision.

Examples of paraphrases learned by NEPAL in these 15 iterations are: erro/engano, erro/falha, ensino/educação, apontar/mostar, alugar/arrendar and questão/assunto.

The evaluation of the three versions of the Promoter was performed by a native speaker of Brazilian Portuguese who was told to evaluate all beliefs promoted by each version. The results of this evaluation are presented in Table 2.

Table 2. Evaluation of beliefs produced by the NEPAL's Promoters

Promoter	Correct	Incorrect
Promoter-0	74.46% (137/184)	25.54% (47/184)
Promoter-1	77.44% (127/164)	22.56% (37/164)
Promoter-2	86.36% (133/154)	13.64% (21/154)

Based on the results in Table 2, it is possible to see a growing trend in the quality of the beliefs: accuracy goes from 74.46% in Promotor-0 to 86.36% in Promotor-2. Over time, NEPAL improved its ability to identify paraphrases, characteristic that demonstrates the never-ending learning potential of NEPAL. Furthermore, NEPAL seems to be robust in relation to its coverage since the percentage of instances that have been promoted to beliefs regarding the number of instances generated remained around 47%.[12]

5 CONCLUSION AND FUTURE WORK

Although the experiment described encompass only 15 iterations, it is possible to note that the paraphrases learned in previous iterations were useful for learning new ones, since the percentage of candidates correctly promoted to beliefs only increased.

However, the current version of NEPAL has several limitations which will be addressed in future work. One of these limitations is related to the incorrect generation of paraphrase candidates involving a word that is part of a bigger multiword expression. To address this limitation, one possibility is to apply a tool capable of dealing with multiword expressions such as the mwetoolkit [20].

[12] 184 out of 398 instances generated during iterations 1-5 were promoted to beliefs by Promotor-0 (46.23%); 164 out of 349 instances generated during iterations 6-10 were promoted to beliefs by Promotor-1 (46.99%); and 154 out of 322 instances generated during iterations 11-15 were promoted to beliefs by the Promotor-2 (47.83%).

References

1. Aziz, W., Specia, L.: Multilingual WSD-like constraints for paraphrase extraction. In: Proceedings of the Seventeenth CoNLL. pp. 202–211. Sofia, Bulgaria (2013)
2. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proceedings of ACL 2005. pp. 597–604. Ann Arbor, USA (2005)
3. Barzilay, R., Elhadad, N., Mckeown, K.: Inferring strategies for sentence ordering in multidocument news summarization. Artificial Intelligence Research 17(2), 35–55 (2002)
4. Barzilay, R., Mckeown, K.: Extracting paraphrases from a parallel corpus. In: Proceedings of ACL 2001. pp. 50–57. Pittsburg, PA, USA (2001)
5. Betteridge, J., Carlson, A., Hong, S.A., Jr, E.H., Law, E.L., Mitchell, T., Wang, S.H.: Toward never ending language learning. In: AAAI Spring Symposium: Learning by Reading and Learning to Read. pp. 1–2 (2009)
6. Brown, P., Pietra, S., Pietra, V., Mercer, R.: The mathematics of machine translation: Parameter estimation. Computational Linguistics 19(2), 263–311 (1993)
7. C. Callison-Burch, P. Koehn, P., Osborne, M.: Improved statistical machine translation using paraphrases. In: Proceedings of the HLT-NAACL. pp. 17–24. New York, NY, USA (2006)
8. Carletta, J.: Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics 22(2), 249–254 (1996)
9. Caseli, H.M.: Indução de léxicos bilíngues e regras para a tradução automática. Ph.D. thesis, ICMC-USP, São Carlos, SP, Brasil (2007)
10. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3) (2011)
11. Ganitkevitch, J., Durme, B.V., Callison-Burch, C.: Ppdb: The paraphrase database. In: Proceedings of the HLT-NAACL. pp. 758–764. Atlanta, USA (2013)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: An update. SIGKDD Explorations 11(1) (2009)
13. Hsu, W.J., Du, M.W.: New algorithms for the lcs problem. Computer and System Sciences 29(2), 133–152 (1984)
14. Ibrahim, A., Katz, B., Lin, J.: Extracting structural paraphrases from aligned monolingual corpora. In: Proceedings of the 2nd International Workshop on Paraphrasing. pp. 57–64. Sapporo, Japan (2003)
15. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the Demo and Poster Sessions of ACL 2007. pp. 177–180. Prague, Czech Republic (2007)
16. Lewis, D.D.: Representation and learning in information retrieval. Ph.D. thesis, Amherst, USA (1992)
17. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29 (2003)
18. Plas, L.V.D., Tiedemann, J.: Finding synonyms using automatic word alignment and measures of distributional similarity. In: Proceedings of the COLING/ACL. pp. 866–873. Sydney, Australia (2006)
19. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, USA (1993)
20. Ramisch, C.: Multiword Expressions Acquisition: A Generic and Open Framework. Theory and Applications of Natural Language Processing series XIV, Springer (2015), iSBN 978-3-319-09206-5
21. Seno, E.R.M., das Graças Volpe Nunes, M.: Reconhecimento de informaes comuns para a fuso de sentenas comparveis do portugułs. Linguamtica (1), 71–87 (2009)