Detection and Annotation of Homographic Puns in intended Humorous Context

Divya Agrawal
MTech Scholar
BIT-Durg

Dr. Ani Thomas
Prof. & HOD(IT)
BIT-Durg

riyaagrawal.bhilai@gmail.comani.thomas@bitdurg.ac.in

Abstract

Word sense disambiguation (WSD) is the chore of identifying a word's meaning in linguistic. Conventional approaches to automatic word sense disambiguation (WSD) rest on the assumption that there exists only one, unambiguous communicative intention underlying every word in a document. However, writers sometimes designate a word to be interpreted as simultaneously carrying many distinct meanings.

Lexical ambiguity, a fundamental characteristic of human linguistics, has long been regarded as a major challenge to machine translation, human–computer interaction, and other applications of computational natural language processing (NLP). Deliberate use of lexical ambiguity also known to be "paronomasia" or "punning" is particularly a common source of humour, and therefore has important implications for how NLP systems process documents and interact with users. The pun or paronomasia is a form of wordplay having multiple meanings of a term, or of a similar sounding words, for an intended humorous or rhetorical effect.

This paper emphasize on the task of classifying a given context into two binary categories: puns and non-puns, concerning itself with finding the word producing the punning effect and annotation of puns with ambiguity in the given context.

Keywords: polysemy, lexical ambiguity, paronomasia, lexicographers, macaroni

1. Introduction

Polysemy or ambiguity is a fundamental characteristic of all natural languages. Lexicographers have recognized that words have multiple meanings, and moreover that more frequently used words have disproportionately more senses than less frequent ones (Zipf 1949). Despite this, humans do not normally perceive any lexical ambiguity in processing written or spoken language. Since computers have no inherent ability to process natural language, the issue of polysemy has been the subject of extensive study in computational linguistics. Word ambiguity has been a major challenge to machine translation, and subsequent researchers have noted its implications for accurate information retrieval, information extraction, and other applications (Agirre& Edmonds 2006).

Ambiguity linguistics consists of a class of language known as *paronomasia*, or *puns*, in which homographic (i.e. coarse-grained) lexical-semantic ambiguity is a deliberateeffect of the communication act. This means the word has two or more separate meanings and the writer is intended of any one of them.

Paronomasia or puns brings a humorous effect by forming a wordplay where a word is used in such a format that fits into all the possible meanings of that specific word. Paronomasia are also a standard rhetorical and poetic device in literature, speeches, slogans, and storytelling, where they can also be used non-humorously.

Puns are classified in following categories:

- Homographic puns- are words that have more than one meaning, despite being spelled identically or have different meanings.
  Examples:
  A horse is a very ***stable*** animal.
  I've been to the dentist many times so I know the ***drill***.
- Homophonic puns- rely on words that sound alike or have similar sounds, rather than a single word with multiple meanings.
  Example:
  Seven days without laughter makes one ***weak***.
- Homonymic puns- these words include both homographs and homophones.
  Example:
  I cried when I found out my ***macaroni*** had expired. It ***pasta*** way.
- Compound puns- that which contains two or more puns in the same sentence.
  Example:
  Why can a man never ***starve*** in the Great Desert? Because he can eat the ***sand which*** is there.
- Recursive puns- the second characteristic of the pun relies on the understanding of the first.
  Example:
  A Freudian slip is when you say one thing but mean ***your mother***(or another).

2. Literature Review

Previous studies on computational detection and comprehension of puns focuses on phonological and syntactic features. Yokogawa (2002), for example, describes a system for detecting the presence of puns in Japanese text, but it works only with puns which are both imperfect and ungrammatical, relying on syntactic cues rather than lexical-semantic information

In similar fashion, Taylor and Mazlack (2004) describe an *n*-gram-based approach for recognizingimperfect puns that are used for humorous effect in English knock-knock jokes. Their focus is on imperfect puns and their use of a fixed syntactic context makes their approach largely inapplicable to arbitrary puns in running text. But for the fact that they are incapable of assigning multiple distinct meanings to the same target, word sense disambiguation algorithms could provide the lexical-semantic understanding necessary to process puns in arbitrary syntactic contexts.

Many approaches to WSD involve evaluating scores for all possible senses of a target word, and then selecting the single highest-scoring one as the “correct” sense. This was attained by having a system that selects the *two* top-scoring senses, one for each meaning of the pun. Because the polysemy exploited by puns is coarse-grained, this naive approach would be inappropriate when the two top-scoring senses are closely related. To account for such cases, it would be helpful to adopt an additional restriction that the second sense selected should have some minimum semantic distance (Budanitsky&Hirst 2006) from the first.

A similar approach could be used for pun detection is by running it through a high-precision WSD system and make a note of the differences in scores between the top two or three semantically dissimilar sense candidates. For unambiguous targets, we would expect the score for the top-chosen sense to greatly exceed those of the others. For puns, however, we would expect the two top-scoring dissimilar candidates to have similar scores, and the third dissimilar sense (if one exists) to score much lower. Given sufficient training data, it may be possible to empirically determine the best score difference thresholds for discriminating puns from non-puns.

In traditional WSD,evaluations are carried out by running the disambiguation system on a large corpus whose target words have been manually annotated by human judges called "gold standard". When the system and gold-standard assignments consist of a single sense each, the exact-match is found and the system receives a score of 1 if it choose the sense specified by the gold standard, and 0 otherwise. Where the system selects a single sense for an instance for which there is more than one correct gold standard sense, the multiple tags are interpreted disjunctively – that is, the system receives a score of 1 if it choose any one of the gold-standard senses(Palmer et al. 2006).

This traditional approach to scoring is not usable asis for pun disambiguation because each pun carries two disjoint but valid sets of sense annotations. Instead, assuming the system selects exactly one sense for each sense set, we would count this as a match (scoring 1) only if each chosen sense can be found in one of the gold-standard sense sets, and no two gold-standard sense sets contain the same chosen sense.

3. Methodology

Homographic puns or paronomasia have identical spelling therefore interpretation of literal and non-literal meaning is dependent on the context information. Thus detection of homographic pun requires a dataset consisting of all the puns along with its meaning which can be used as pun dictionary. Context information is a crucial factor in detecting homographic puns.

This system is designed for sentences consisting of only one word homographic puns and with two or more meanings. A sentence containing the pun will act as input for the system and a parts of speech tagger is used to extract noun and verbs dropping the rest of the contents of the sentence which is further passed to a stemming section where the root words were extracted. A large interface database is used called as generating pun dictionary. Now the aim will be of finding all those word having two or more meanings with the similar spelling representation which will be created with the help of WordNet dictionary. This leads to a binary classification of any word to be identified as pun or non-pun. After passing the extracted words through this pun dictionary the different meanings of the pun associated with the other reference word in the sentence could be found. Also when the pun is detected another task is to find the location at which the pun is present in the sentence.

A pun word will have associated word with the meaning. The associated word will help in the annotation of puns, identifying sense of humour and usage in that context. Since NLP contains word sense disambiguation, the greater the number of associated words,greater will be the different meanings and the higher accuracy will be obtained. The words with the low frequency may contribute for annotation of the nearest meaning.

4. Conclusion

The detection and identification of puns are the recent important research topics in Natural Language Processing (NLP). Homographic puns are generally figurative in nature because of their spelling resemblance. An attempt is made to develop pun dictionary with the help of WordNet dictionary available online and then an automatedsystem to detect puns: whether it is present in input sentence or not by passing the tokens of the sentence to the created dictionary and to find the location of the present pun.

5. References

[1] Agirre, E. & Edmonds, P. (eds.) (2006). *Word Sense Disambiguation: Algorithms and Applications*. Text, Speech, and Language Technology, Volume 33. Berlin: Springer.

[2] Budanitsky, A. &Hirst, G. (2006). 'Evaluating WordNet-based measures of lexical semantic relatedness'. *Computational Linguistics* 32 (1), pp. 13–47.

[3] Hirst, G. (1987). *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge: Cambridge University Press.

[4] Palmer, M., Ng, H. T. & Dang, H. T. (2006). 'Evaluation of WSD systems', in Agirre, E. & Edmonds, P. (eds.), *Word Sense Disambiguation: Algorithms and Applications*. Text, Speech, and Language Technology, Volume 33. Berlin: Springer.

[5] Taylor, J. M. &Mazlack, L. J. (2004). 'Computationally recognizing wordplay in jokes', in Forbus, K., Gentner, D. &Regier, T. (eds.), *Proceedings of the 26th Annual Conference of the Cognitive Science Society (CogSci 2004)*, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 1315–1320.

[6] Yokogawa, T. (2002). 'Japanese pun analyzer using articulation similarities', in *Proceedings of the 11th IEEE International Conference on Fuzzy Systems (FUZZ 2002).* Volume 2. Piscataway, NJ: IEEE Press, pp. 1114–1119.

[7] Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort*. Cambridge, MA: Addison–Wesley.

[8] Tristan Miller and MladenTurkovic, "Towards the automatic detection and identification of English puns", European Journal of Humour Research 4 (1) 59–75.

[9] Samuel Doogan, Aniruddha Ghosh,Hanyang Chen, Tony Veale, "Idiom Savant at Semeval-2017 Task 7: Detection and Interpretation ofEnglish Puns".

[10] Tristan Miller and IrynaGurevych, " Automatic disambiguation of English puns", Proceedings of the 53rd Annual Meeting of the Association for Computational Linguisticsand the 7th International Joint Conference on Natural Language Processing, pages 719–729,Beijing, China, July 26-31, 2015. c 2015 Association for Computational Linguistics.

[11] http://examples.yourdictionary.com/examples-of-puns.html#rme4H1ivlB2I1RJL.99, Examples of puns

[12] https://en.wikipedia.org/wiki/Pun, Pun Wikipedia

[13] http://english.tutorvista.com/literary-response/pun.html , TutorVista.com

[14] http://www.literarydevices.com/pun/ , Literacy Devices.

[15] https://www.buzzle.com/articles/pun-examples.html , Buzzle.com, "These Hilarious Pun Examples Will Make You Laugh Like Crazy".