http://dx.doi.org/10.7592/EJHR2016.4.1.miller

European Journal of Humour Research 4 (1) 59–75
www.europeanjournalofhumour.org

Towards the automatic detection and identification of English puns

Tristan Miller
Ubiquitous Knowledge Processing Lab, Department of Computer Science, Technische Universität Darmstadt, Germany
miller@ukp.informatik.tu-darmstadt.de

Mladen Turković[1]
J.E.M.I.T. d.o.o., Pula, Croatia
mladen.turkovic@massine.com

Abstract

Lexical polysemy, a fundamental characteristic of all human languages, has long been regarded as a major challenge to machine translation, human–computer interaction, and other applications of computational natural language processing (NLP). Traditional approaches to automatic word sense disambiguation (WSD) rest on the assumption that there exists a single, unambiguous communicative intention underlying every word in a document. However, writers sometimes intend for a word to be interpreted as simultaneously carrying multiple distinct meanings. This deliberate use of lexical ambiguity — i.e. punning — is a particularly common source of humour, and therefore has important implications for how NLP systems process documents and interact with users. In this paper we make a case for research into computational methods for the detection of puns in running text and for the isolation of the intended meanings. We discuss the challenges involved in adapting principles and techniques from WSD to humorously ambiguous text, and outline our plans for evaluating WSD-inspired systems in a dedicated pun identification task. We describe the compilation of a large manually annotated corpus of puns and present an analysis of its properties. While our work is principally concerned with simple puns which are monolexemic and homographic (i.e. exploiting single words which have different meanings but are spelled identically), we touch on the challenges involved in processing other types.

Keywords: puns, word sense disambiguation, lexical semantics, paronomasia, sense ambiguity.

1. Introduction

Polysemy is a fundamental characteristic of all natural languages. Writers, philosophers, linguists, and lexicographers have long recognised 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; each polysemous word is unconsciously and automatically understood to mean exactly what the writer or speaker intended (Hirst 1987). Computers, however, have no inherent ability to process natural language, and the issue of polysemy has been the subject of extensive study in computational linguistics since the very nascence of the field. Computing pioneers in the 1940s recognised polysemy as a major challenge to machine translation, and subsequent researchers have noted its implications for accurate information retrieval, information extraction, and other applications. There is by now a considerable body of research on the problem of *word sense disambiguation* — that is, having a computer automatically identify the correct sense for a word in a given context (Agirre & Edmonds 2006).

Traditional approaches to word sense disambiguation rest on the assumption that there exists a single unambiguous communicative intention underlying every word in the document or speech act under consideration.[2] However, there exists a class of language constructs known as *paronomasia*, or *puns*, in which homonymic (i.e. coarse-grained) lexical-semantic ambiguity is a *deliberate* effect of the communication act. That is, the writer[3] intends for a certain word or other lexical item to be interpreted as simultaneously carrying two or more separate meanings. Though puns are a recurrent and expected feature in many discourse types, current word sense disambiguation systems, and by extension the higher-level natural language applications making use of them, are completely unable to deal with them.

In this article, we present our arguments for why computational detection and interpretation of puns are important research questions. We discuss the challenges involved in adapting traditional word sense disambiguation techniques to intentionally ambiguous text and outline our plans for evaluating these adaptations in a controlled setting. We also describe in detail our creation of a large data set of manually sense-annotated puns, including the specialised tool we have developed to apply the sense annotations.

2. Background

2.1. Puns

A *pun* is a writer's use of a word in a deliberately ambiguous way, often to draw parallels between two concepts so as to make light of them. They are a common source of humour in jokes and other comedic works; there are even specialised types of jokes, such as the *feghoot* (Ritchie 2004: 223) and *Tom Swifty* (Lippmann & Dunn 2000), in which a pun always occurs in a fixed syntactic or stylistic structure. Puns are also a standard rhetorical and poetic device in literature, speeches, slogans, and oral storytelling, where they can also be used non-humorously. Shakespeare, for example, is famous for his use of puns, which occur with high frequency even in his non-comedic works.[4] Both humorous and non-humorous puns have been the subject of extensive study, which has led to insights into the nature of language-based humour and wordplay, including their role in commerce, entertainment, and health care; how they are

processed in the brain; and how they vary over time and across cultures (e.g., Monnot 1982; Culler 1988; Lagerwerf 2002; Bell et al. 2011; Bekinschtein et al. 2011). Study of literary puns imparts a greater understanding of the cultural or historical context in which the literature was produced, which is often necessary to properly interpret and translate it (Delabastita 1997b).

Humanists have grappled with the precise definition and classification of puns since antiquity. Recent scholarship tends to categorise puns not into a single overarching taxonomy, but rather by using clusters of mutually independent features (Delabastita 1997a). The feature of greatest immediate interest to us is *homography* – that is, whether the words for the two senses of the pun share the same orthographic form. (By contrast, some prior work in computational humour has concerned itself with the criterion of *homophony*, or whether the two words are pronounced the same way. Puns can be homographic, homophonic, both, or neither; those in the last category are commonly known as *imperfect* puns.) Other characteristics of puns important for our work include whether they involve compounds, multiword expressions, or proper names, and whether the pun's multiple meanings involve multiple parts of speech. We elaborate on the significance of these characteristics later in this article.

2.2. Word sense disambiguation

Word sense disambiguation is the task of computationally determining which sense of a polysemous term is the one intended when that term is used in a particular communicative act. Though regarded as one of the most fundamental and difficult of all problems in artificial intelligence, even today's imperfect WSD systems have made measurable and encouraging improvements to higher-level NLP applications such as search engines and machine translation systems. WSD has also been proposed or implemented as a component in tools for information extraction, content analysis, writing assistance, and computational lexicography (Navigli 2009).

Approaches to WSD differ widely in the knowledge sources and strategies they employ, an overview of which can be found in surveys by Agirre and Edmonds (2006) and Navigli (2009). At minimum, though, a WSD system takes three inputs – the *target word* to be disambiguated, the *context* surrounding it, and a *sense inventory* listing all possible senses of the target – and produces as output a list of senses from the inventory which correspond to the target instance. The sense inventory can be a simple machine-readable dictionary or thesaurus listing textual definitions or synonyms for each sense, though nowadays more sophisticated lexical-semantic resources (LSRs) are often used instead of or alongside these. The LSR most commonly used in English-language WSD research is WordNet (Fellbaum 1998), an electronic lexical database which, in addition to providing definitions and example sentences, links words and concepts into a network by means of lexical and semantic relations such as derivation, hypernymy, and meronymy.

An implicit assumption made by all WSD algorithms heretofore engineered is that the targets are used more or less unambiguously. That is, while the sense inventory may give a multiplicity of senses for a word, at most one of them (or perhaps a small cluster of closely related senses) is correct when that word is used in a particular context. Where a WSD system does select multiple sense annotations for a given target, this is taken to mean that the target has a single coarse-grained meaning that subsumes those senses, or that the distinction between them is unimportant. The assumption of unambiguity covers not only semantics but also syntax: it is assumed that each target has a single part of speech and lemma (i.e. canonical form) which are known *a priori* or can be deduced with high accuracy.

3. Motivation and previous work

Puns have been discussed in rhetorical and literary criticism since ancient times, and in recent years have increasingly come to be seen as a respectable research topic in traditional linguistics and the cognitive sciences (Delabastita 1997a). It is therefore surprising that they have attracted very little attention in the fields of computational linguistics and natural language processing. What little research has been done is confined largely to computational mechanisms for pun generation (in the context of natural language generation for computational humour) and to computational analysis of phonological properties of puns (e.g. Binsted & Ritchie 1994, 1997; Hempelmann 2003a, 2003b; Ritchie 2005; Hong & Ong 2009; Kawahara 2010). A fundamental task which has not yet been as widely studied is the automatic detection and identification of intentional lexical ambiguity – that is, given a text, does it contain any lexical items which are used in a deliberately ambiguous manner, and if so, what are the intended meanings?

We consider these to be important research questions with a number of real-world applications. For example:

Human–computer interaction. It has often been argued that humour can enhance human–computer interaction (HCI) (Hempelmann 2008), and at least one study has already shown that incorporating canned humour into a user interface can increase user satisfaction without adversely affecting user efficiency (Morkes et al. 1999). Interestingly, the same study found that some users of the humorous interface told jokes of their own to the computer. We posit that having the computer recognise a user's punning joke and produce a contextually appropriate response (which could be as simple as canned laughter or as complex as generating a similar punning joke in reciprocation) could further enhance the HCI experience.

Sentiment analysis. Sentiment analysis is a form of automated text analysis that seeks to identify subjective information in source materials. It holds particular promise in fields such as market research, where it is useful to track a population's attitude towards a certain person, product, practice, or belief, and to survey how people and organizations try to influence others' attitudes. As it happens, puns are particularly common in advertising, where they are used not only to create humour but also to induce in the audience a valenced attitude toward the target (Monnot 1982; Valitutti et al. 2008). (This attitude need not be positive – a commercial advertisement could use unflattering puns to ridicule a competitor's product, and a public service announcement could use them to discommend undesirable behaviour.) Recognising instances of such lexical ambiguity and understanding their affective connotations would be of benefit to systems performing sentiment analysis on persuasive texts.

Machine-assisted translation. Some of today's most widely disseminated and translated popular discourses – particularly television shows and movies – feature puns and other forms of wordplay as a recurrent and expected feature (Schröter 2005). Puns pose particular challenges for translators, who need not only to recognise and comprehend each instance of humour-provoking ambiguity, but also to select and implement an appropriate translation strategy. Future NLP systems could assist translators in flagging intentionally ambiguous words for special attention, and where they are not directly translatable (as is usually the case), the systems may be able to propose ambiguity-preserving alternatives which best match the original pun's double meaning.

Digital humanities. Wordplay is a perennial topic of scholarship in literary criticism and analysis. Shakespeare's puns, for example, are one of the most intensively studied aspects of his rhetoric, with countless articles and even entire books (Wurth 1895; Rubinstein 1984; Keller 2009) having been dedicated to their enumeration and analysis. It is not hard to image how computer-assisted detection, classification, and analysis of puns could help scholars in the digital humanities in producing similar surveys of other œuvres.

It would seem that an understanding of lexical semantics is necessary for any implementation of the above-noted applications. However, the only previous studies on computational detection and comprehension of puns that we are aware of focus 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 a somewhat similar vein, Taylor and Mazlack (2004) describe an *n*-gram-based approach for recognising when imperfect puns are used for humorous effect in a very narrow class of English knock-knock jokes. Their focus 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. (We are not, in fact, the first to suggest this – Mihalcea and Strapparava [2006] have also speculated that semantic analysis, such as via word sense disambiguation or domain disambiguation, could aid in the detection of humorous incongruity and opposition.) In the following section, we sketch some ideas of how traditional WSD systems could be adapted to recognise and sense-annotate puns, and how such adaptations could be evaluated in a controlled setting.

4. Pun detection and disambiguation

4.1. Algorithms

Computational processing of puns involves two separate tasks: In *pun detection*, the object is to determine whether or not a given context contains a pun, or more precisely whether any given word in a context is a pun. In *pun identification* (or *pun disambiguation*), the object is to identify the two meanings of a term previously detected, or simply known *a priori*, to be a pun.

To understand how traditional word sense disambiguation approaches can be adapted to the latter task, recall that they work by attempting to assign a single sense to a given target. If they fail to make an assignment, this is generally for one of the following reasons:

1. The target word does not exist in the sense inventory.
2. The knowledge sources available to the algorithm (including the context and information provided by the sense inventory) are insufficient to link any one candidate sense to the target.
3. The sense information provided by the sense inventory is too fine-grained to distinguish between closely related senses.
4. The target word is used in an intentionally ambiguous manner, leading to indecision between coarsely related or unrelated senses.

We hold that for this last scenario, a disambiguator's inability to discriminate senses should not be seen as a failure condition, but rather as a limitation of the WSD task as traditionally defined. By reframing the task so as to permit the assignment of multiple senses (or groups thereof), we can allow disambiguation systems to sense-annotate intentionally ambiguous constructions such as puns.

Many approaches to WSD involve computing some score for all possible senses of a target word, and then selecting the single highest-scoring one as the "correct" sense. The most straightforward modification of these techniques to pun disambiguation, then, is to have the systems select 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 the requisite problem of pun detection: to determine whether or not a given word is a pun, run 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. (We hasten to note, however, that such an approach would not be able to distinguish between intentional and accidental puns. Whether this is a limitation or a feature would depend on the ultimate application of the pun detection system.)

4.2. Evaluation

In traditional WSD, *in vitro* evaluations are conducted by running the disambiguation system on a large "gold standard" corpus whose target words have been manually annotated by human judges. For the case that the system and gold-standard assignments consist of a single sense each, the exact-match criterion is used: the system receives a score of 1 if it chose 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 chose any one of the gold-standard senses, and 0 otherwise. Overall performance is reported in terms of *precision* (the sum of scores divided by the number of attempted targets) and *recall* or *accuracy* (the sum of scores divided by the total number of targets) (Palmer et al. 2006).

This traditional approach to scoring is not usable as-is 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.

By contrast, the task of pun detection is straightforward to evaluate. Here the system annotates each word (or context, for the coarser-grained variant of the task) as either containing or not containing a pun. Each case where the system and human annotators agree nets the system a score of 1. Overall performance would be reported in terms of precision and recall, as above.

5. Data set

As in traditional word sense disambiguation, a prerequisite for pun disambiguation is a corpus of positive examples where human annotators have already identified the ambiguous words and marked up their various meanings with reference to a given sense inventory. For pun detection, a corpus of negative examples is also required. In this section we briefly review the data sets which have been used in past work and describe the creation of our own.

5.1. Previous resources

There are a number of English-language pun corpora which have been used in past work, usually in linguistics or the social sciences. In their work on computer-generated humour, Lessard et al. (2002) use a corpus of 374 Tom Swifties taken from the Internet, plus a well-balanced corpus of 50 humorous and non-humorous lexical ambiguities generated programmatically (Venour 1999). Hong and Ong (2009) also study humour in natural language generation, using a smaller corpus of 27 punning riddles derived from a mix of natural and artificial sources. In their study of wordplay in religious advertising, Bell et al. (2011) compile a corpus of 373 puns taken from church marquees and literature, and compare it against a general corpus of 1515 puns drawn from Internet websites and a specialised dictionary (Crosbie 1977). Zwicky and Zwicky (1986) conduct a phonological analysis on a corpus of "several thousand" puns, some of which they collected themselves from advertisements and catalogues, and the remainder of which were taken from previously published collections (Crosbie 1977; Monnot 1981; Sharp 1984). Two studies on cognitive strategies used by second language learners (Kaplan & Lucas 2001; Lucas 2004) used a corpus of 58 jokes compiled from newspaper comics, 32 of which rely on lexical ambiguity. Bucaria (2004) conducts a linguistic analysis of a corpus of 135 humorous newspaper headlines, about half of which exploit lexical ambiguity.

Such corpora – particularly the larger ones – are good evidence that intentionally lexical ambiguous exemplars exist in sufficient numbers to make a rigorous evaluation of our proposed tasks feasible. Unfortunately, none of the above-mentioned corpora have been published in full, and none of them are systematically sense-annotated. This has motivated us to produce our own corpus of puns, the construction and analysis of which is described in the following subsection.

5.2. Raw data

Our aim was to collect approximately 2000 puns in short contexts, as this number of instances is typical of testing data sets used in past WSD competitions such as Senseval and SemEval (Palmer et al. 2001; Kilgarriff 2001; Snyder & Palmer 2004; Navigli et al. 2007). To keep the complexity of our disambiguation method and of our evaluation metrics manageable in this pilot study, we decided to consider only those examples meeting the following four criteria:

One pun per instance: Of all the lexical units in the instance, one and only one may be a pun. Adhering to this restriction makes pun detection within contexts a binary classification task, which simplifies evaluation and leaves the door open for use of certain machine learning algorithms.

One content word per pun: The lexical unit forming the pun must consist of, or contain, only a single content word (i.e. a noun, verb, adjective, or adverb), excepting adverbial particles of

phrasal verbs. (For example, a pun on "car" is acceptable because it is a single content word, whereas a pun on "to" is not because it is not a content word. A pun on "ice cream" is unacceptable, because although it is a single lexical unit, it consists of two content words. A pun on the phrasal verb "put up with" meets our criteria: although it has three words, only one of them is a content word.) This criterion is important because, in our observations, it is usually only one word which carries ambiguity in puns on compounds and multi-word expressions. Processing these cases would require the annotator (whether human or machine) to laboriously partition the pun into (possibly overlapping) sense-bearing units and to assign sense sets to each of them.

Two meanings per pun: The pun must have exactly two distinct meanings. Though sources tend to agree that puns have only two senses (Redfern 1984; Attardo 1994), our annotators identified a handful of examples where the pun could plausibly be analyzed as carrying three distinct meanings. To simplify our manual annotation procedure and our evaluation metrics we excluded these rare outliers from our corpus.

Weak homography: While the WSD approaches we plan to evaluate would probably work for both homographic and heterographic puns, admitting the latter would require the use of pronunciation dictionaries and application of phonological theories of punning in order to recover the target lemmas (Hempelmann 2003a). As our research interests are in lexical semantics rather than phonology, we focus for the time being on puns which are more or less homographic. More precisely, we stipulate that the lexical units corresponding to the two distinct meanings must be spelled exactly the same way, with the exception that particles and inflections may be disregarded. This somewhat softer definition of homography allows us to admit a good many morphologically interesting cases which were nonetheless readily recognised by our human annotators.

We began by pooling together some of the previously mentioned data sets, original pun collections made available to us by professional humorists, and freely available pun collections from the Web. After filtering out duplicates, these amounted to 7750 candidate instances, mostly in the form of short sentences. About half of these come from the Pun of the Day website, a quarter from the personal archive of author Stan Kegel, and the remainder from various private and published collections. We then employed human annotators to filter out all instances not meeting the above-noted criteria; this winnowed the collection down to 1652 positive instances. These range in length from 3 to 44 words, with an average length of 11.8.

For our corpus of negative examples, we followed Mihalcea and Strapparava (2005, 2006) and assembled a raw database of 2972 proverbs and famous sayings from various Web sources. These are similar in length and style to our positive examples; many of them contain humour but few of them contain puns. At the time of writing we are still in the process of filtering these. However, based on our work so far, we estimate about two thirds of them to contain a word which is used as a pun in the confirmed positive examples.

5.3. Sense annotation

Manual linguistic annotation, and sense annotation in particular, is known to be a particularly arduous and expensive task (Mihalcea & Chklovski 2003). The process can be sped up somewhat through the use of dedicated annotation support software. However, existing sense

annotation tools, such as Stamp (Hovy et al. 2006), SATANiC (Passonneau et al. 2012), and WebAnno (Yimam et al. 2013), and the annotated corpus formats they write, do not support specification of distinct groups of senses per instance. It was therefore necessary for us to develop our own sense annotation tool, along with a custom Senseval-inspired corpus format.

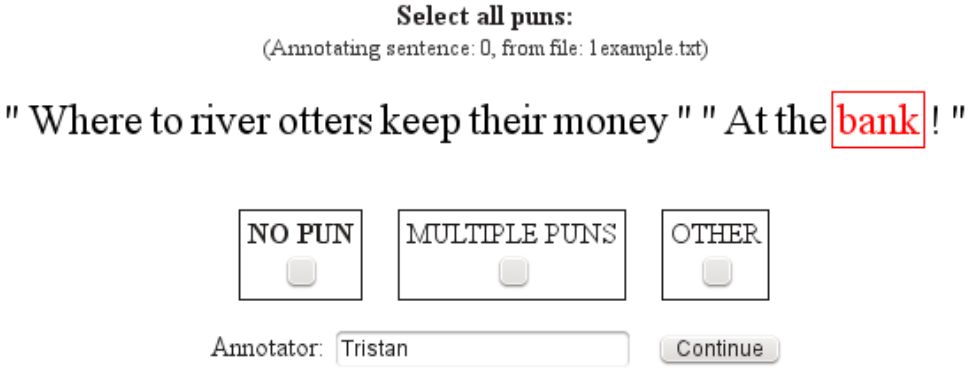

Figure 1. Selecting pun words in Punnotator.

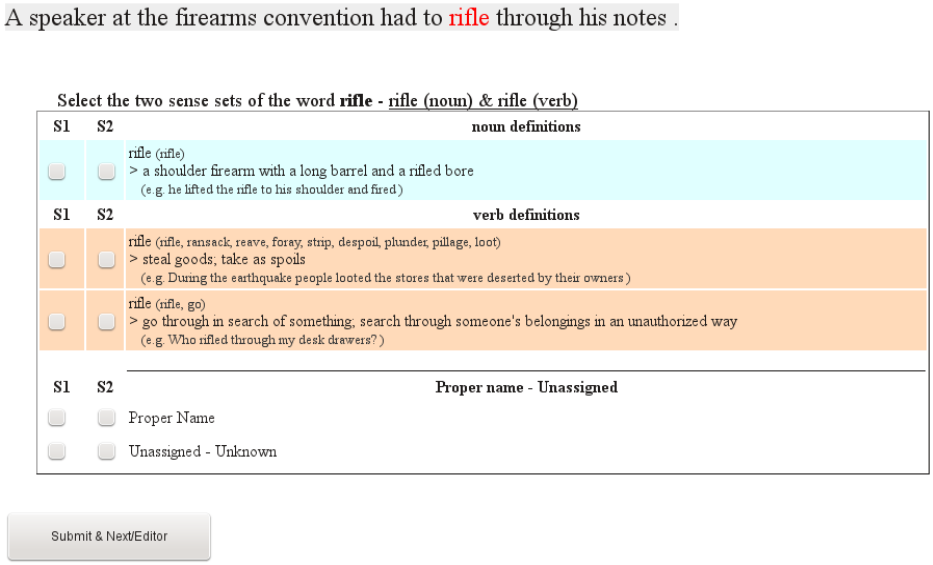

Figure 2. Selecting definitions in Punnotator.

Our annotation tool, Punnotator, runs as a Web application on a PHP-enabled server. It reads in a simple text file containing the corpus of instances to annotate and presents them to the user one at a time through their web browser. For each instance, the user is asked to select the pun's content word, or else to check one of several boxes in the event that the instance has no pun or is otherwise invalid. (See Figure 1.) Punnotator then determines all possible lemmas of the selected content word, retrieves their definitions from a sense inventory, and presents them to the user in a table. (See Figure 2.) Unlike with traditional sense annotation tools, definitions from all parts of speech are provided, since puns often cross parts of speech.

The definition table includes two columns of checkboxes representing the two distinct meanings for the pun. In each column, the user checks all those definitions which correspond to one of the pun's two meanings. It is possible to select multiple definitions per column, which indicates that the user believes them to be indistinguishable or equally applicable for the intended meaning. The only restriction is that the same definition may not be checked in both columns. Following Senseval practice, if one or both of the meanings of the pun are not represented by any of the listed definitions, the user may check one of two special checkboxes at the bottom of the list to indicate that the meaning is a proper name or otherwise missing from the sense inventory.

We elected to use the latest version of WordNet (3.1) as the sense inventory for our annotations. Though WordNet has often been criticised for the overly fine granularity of its sense distinctions (Ide & Wilks 2006), it has the advantage of being freely available, of being the *de facto* standard LSR for use in WSD evaluations, and of being accessible through a number of flexible and freely available software libraries.

5.4. Analysis

Two trained judges used our Punnotator tool to manually sense-annotate all 1652 positive instances. They agreed on which word was the pun in 1634 cases, a raw agreement of 98.91 per cent. For these 1634 cases, we measured inter-annotator agreement on the sense assignments using Krippendorff's α (Krippendorff 1980). This is a chance-correcting metric ranging in (−1,1], where 1 indicates perfect agreement, −1 perfect disagreement, and 0 the expected score for random labelling. Our distance metric for α is a straightforward adaptation of the MASI set comparison metric (Passonneau 2006). Whereas standard MASI, $d_M(A,B)$, compares two annotation sets *A* and *B*, our annotations take the form of unordered *pairs* of sets $\{A_1, A_2\}$ and $\{B_1, B_2\}$. We therefore find the mapping between elements of the two pairs that gives the lowest total distance, and halve it:

$$d_{M'}(\{A_1, A_2\}, \{B_1, B_2\}) = \frac{1}{2}\min\big(d_M(A_1, B_1) + d_M(A_2, B_2), d_M(A_1, B_2) + d_M(A_2, B_1)\big).$$

With this method we observe a Krippendorff's α of 0.777; this is only slightly below the 0.8 threshold recommended by Krippendorff, and far higher than what has been reported in other sense annotation studies (Passonneau et al. 2006; Jurgens & Klapaftis 2013).

Where possible, we resolved sense annotation disagreements automatically by taking the intersection of corresponding sense sets. For cases where the annotators' sense sets were disjoint or contradictory (including the cases where the annotators disagreed on the pun word), we had an independent human adjudicator attempt to resolve the disagreement in favour of one annotator or the other. This left us with 1607 instances; pending clearance of the distribution rights, we will make some or all of this annotated data set available on our website at https://www.ukp.tu-darmstadt.de/data/. Following are our observations on the qualities of the annotated corpus:

Sense coverage. Of the 1607 instances in the corpus, the annotators were able to successfully annotate both sense sets for 1298 (80.8 per cent). For 303 instances (18.9 per cent), WordNet was found to lack entries for only one of the sense sets, and for the remaining 6 instances (0.4 per cent), WordNet lacked entries for both sense sets. By comparison, in the Senseval and SemEval corpora the proportion of target words with unknown or unassignable senses ranges from 1.7 to 6.8 per cent. This difference can probably be explained by the differences in genre: WordNet was constructed by annotating a subset of the Brown Corpus, a million-word corpus of American texts published in 1961 (Miller et al. 1993). The Brown

Corpus samples a range of genres, including journalism and technical writing, but not joke books. The Senseval and SemEval data sets tend to use the same sort of news and technical articles found in the Brown Corpus, so it is not surprising that a greater proportion of their words' senses can be found in WordNet.

Our 2596 successfully annotated sense sets have anywhere from one to seven senses each, with an average of 1.08. As expected, then, WordNet's sense granularity proved to be somewhat finer than necessary to distinguish between the senses in our data set, though only marginally so.

Part of speech distribution. Of the 2596 successfully annotated sense sets, 50.2 per cent contain noun senses only, 33.8 per cent verb senses only, 13.1 per cent adjective senses only, and 1.6 per cent adverb senses only. As previously noted, however, the semantics of puns sometimes transcends part of speech: 1.3 per cent of our individual sense sets contain some combination of senses representing two or three different parts of speech, and of the 1298 instances where both meanings were successfully annotated, 297 (22.9 per cent) have sense sets of differing parts of speech (or combinations thereof). This finding confirms the concerns we raised in Section 2.2. that pun disambiguators, unlike traditional WSD systems, cannot rely on the output of a part-of-speech tagger to narrow down the list of sense candidates.

Polysemy. Because puns have no fixed part of speech, each target term in the data set can have more than one "correct" lemma. An automatic pun disambiguator must therefore consider all possible senses of all possible lemmas of a given target. The annotated senses for each target in our data set represent anywhere from one to four different lemmas (without distinction of part of speech), with a mean of 1.2. The number of candidate senses associated with these lemma sets ranges from 1 to 79, with a mean of 12.4.

Of course, a real-world pun disambiguator will not know *a priori* which lemmas are the correct ones for a given target in a given context. On our data set such a system must select lemmas and senses from a significantly larger pool of candidates (on average 1.5 lemmas and 14.2 senses per target). Recall that on average, only 1.08 of these senses are annotated as "correct" in any given sense set.

Target location. During the annotation process it became obvious that the vast majority of puns were located towards the end of the context. As this sort of information could prove helpful to a disambiguation system, we calculated the frequency of target words occurring in the first, second, third, and fourth quarters of the contexts. As predicted, we found that the final quarter of the context is the overwhelmingly preferred pun location (82.8 per cent of instances), followed distantly by the third (9.3 per cent), second (6.7 per cent), and first (1.2 per cent). This observation accords with previous empirical studies of large joke corpora, which found that the punchline occurs in a terminal position more than 95 per cent of the time (Attardo 1994: ch. 2).

6. Conclusion and future work

In this paper, we have advanced some arguments for why computational detection and understanding of puns are worthy research topics, pointing to potential applications in text analysis, human–computer interaction, and machine translation. We have described in high-level terms how techniques from traditional word sense disambiguation could be adapted to these tasks and how pun detection and disambiguation systems could be evaluated in a controlled

setting. In preparation for such evaluations, we have developed a custom sense annotation tool for puns and used it to construct a large, manually sense-annotated corpus of homographic English puns, for which we have given an overview of selected properties.

At the time of writing we have already begun adapting and evaluating specific WSD algorithms to the task of pun disambiguation. These include naive approaches such as the random sense and most frequent sense baselines (Gale et al. 1992), as well as state-of-the-art systems such as those described by Navigli and Lapata (2010) and Miller et al. (2012). The description and evaluation of some of these adapted systems will be the focus of a forthcoming paper (Miller & Gurevych 2015). Beyond this, our immediate future goals are to complete the construction of our corpus of negative examples and to design, implement, and evaluate various pun detection algorithms. Provided our pun detection and disambiguation systems achieve acceptable levels of accuracy, the next steps would be to incorporate them in the higher-level NLP applications we introduced in Section 3 and to perform *in vivo* evaluations.

Acknowledgements

The work described in this paper was supported by the Volkswagen Foundation as part of the Lichtenberg Professorship Program under grant No. I/82806. The authors thank John Black, Matthew Collins, Don Hauptman, Christian F. Hempelmann, Stan Kegel, Andrew Lamont, Beatrice Santorini, and Andreas Zimpfer for helping us build our data set.

Notes

[1] The work described in this paper was carried out while this author was at the Ubiquitous Knowledge Processing Lab in Darmstadt, Germany.

[2] Under this assumption, lexical ambiguity arises due to there being a plurality of words with the same surface form but different meanings, and the task of the interpreter is to select correctly among them. An alternative view is that each word is a single lexical entry whose specific meaning is *underspecified* until it is activated by the context (Ludlow 1996). In the case of *systematically polysemous* terms (i.e. words which have several related senses shared in a systematic way by a group of similar words), it may not be necessary to disambiguate them at all in order to interpret the communication (Buitelaar 2000). While there has been some research in modelling lexical-semantic underspecification (e.g. Jurgens 2014), these approaches are intended for closely related senses such as those of systematically polysemous terms, not those of coarser-grained homonyms which are the subject of this paper.

[3] Puns can and do, of course, occur in spoken communication as well. Though much of what we cover in this article is equally applicable to written and spoken language, for the purposes of simplification we refer henceforth only to written texts.

[4] Keller (2009) provides frequency lists of rhetorical figures in nine of Shakespeare's plays (four comedies, four tragedies, and one history). Puns, in the sense used in this article, were observed at a rate of 17.4 to 84.7 instances per thousand lines, or 35.5 on average.

References

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

Attardo, S. (1994). *Linguistic Theories of Humor*. Berlin: Mouton de Gruyter.

Bekinschtein, T. A., Davis, M. H., Rodd, J. M., & Owen, A. M. (2011). 'Why clowns taste funny: The relationship between humor and semantic ambiguity'. *The Journal of Neuroscience*, 31 (26), pp. 9665–9671.

Bell, N. D., Crossley, S., & Hempelmann, C. F. (2011). 'Wordplay in church marquees'. *HUMOR: International Journal of Humor Research* 24 (2), pp. 187–202.

Binsted, K. & Ritchie, G. (1994). 'An implemented model of punning riddles', in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994)*, Menlo Park, CA: AAAI Press, pp. 633–638.

Binsted, K. & Ritchie, G. (1997). 'Computational rules for generating punning riddles'. *HUMOR: International Journal of Humor Research* 10 (1), pp. 25–76.

Bucaria, C. (2004). 'Lexical and syntactic ambiguity as a source of humor: The case of newspaper headlines'. *HUMOR: International Journal of Humor Research*, 17 (3), pp. 279–309.

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

Buitelaar, P. (2000). 'Reducing lexical semantic complexity with systematic polysemous classes and underspecification', in *Proceedings of the 2000 NAACL-ANLP Workshop on Syntactic and Semantic Complexity in Natural Language Processing Systems, Volume 1*, Stroudsburg, PA: Association for Computational Linguistics, pp. 14–19.

Crosbie, J. S. (1977). *Crosbie's Dictionary of Puns*. New York: Harmony.

Culler, J. D. (ed.) (1988). *On Puns: The Foundation of Letters*. Oxford: Basil Blackwell.

Delabastita, D. (1997a). 'Introduction', in Delabastita, D. (ed.), *Traductio: Essays on Punning and Translation*, Manchester: St. Jerome, pp. 1–22.

Delabastita, D. (ed.) (1997b). *Traductio: Essays on Punning and Translation*. Manchester: St. Jerome.

Fellbaum, C. (ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Gale, W., Ward Church, K. & Yarowsky, D. (1992). 'Estimating upper and lower bounds on the performance of word-sense disambiguation programs', in *Proceedings of the 30th Annual Meeting of the Association of Computational Linguistics (ACL 1992)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 249–256.

Hempelmann, C. F. (2003a). *Paronomasic Puns: Target Recoverability Towards Automatic Generation*. West Lafayette, IN: Purdue University. PhD thesis.

Hempelmann, C. F. (2003b). 'YPS – The Ynperfect Pun Selector for computational humor', in *Proceedings of the Workshop on Humor Modeling in the Interface at the Conference on Human Factors in Computing Systems (CHI 2003)*, New York: Association for Computing Machinery.

Hempelmann, C. F. (2008). 'Computational humor: Beyond the pun?' In Raskin, V. (ed.), *The Primer of Humor Research*. Humor Research, Volume 8. Berlin: Mouton de Gruyter , pp. 333–360.

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

Hong, B. A. & Ong, E. (2009). 'Automatically extracting word relationships as templates for pun generation', in *Proceedings of the 1st Workshop on Computational Approaches to Linguistic Creativity (CALC 2009)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 24–31.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L. & Weischedel, R. (2006). 'OntoNotes: The 90% solution', in *Proceedings of the Human Language Technology Conference of the NAACL (Short Papers) (HLT-NAACL 2006)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 57–60.

Ide, N. & Wilks, Y. (2006). 'Making sense about sense', in Agirre, E. & Edmonds, P. (eds.), *Word Sense Disambiguation: Algorithms and Applications*. Text, Speech, and Language Technology, Volume 33. Berlin: Springer.

Jurgens, D. (2014). 'An analysis of ambiguity in word sense annotations', in Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J. & Piperidis, S. (eds.), *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, Paris: European Language Resources Association, pp. 3006–3012.

Jurgens, D. & Klapaftis, I. (2013). 'SemEval-2013 Task 13: Word sense induction for graded and non-graded senses', in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 290–299.

Kaplan, N. & Lucas, T. (2001). 'Comprensión del humorismo en inglés: Estudio de las estrategias de inferencia utilizadas por estudiantes avanzados de inglés como lengua extranjera en la interpretación de los retruécanos en historietas cómicas en lengua inglesa'. *Anales de la Universidad Metropolitana* 1 (2), pp. 245–258.

Kawahara, S. (2010). *Papers on Japanese imperfect puns*. Online collection of previously published journal and conference articles. Available online: http://user.keio.ac.jp/~kawahara/pdf/punbook.pdf [Accessed 17 June 2015].

Keller, S. D. (2009). *The Development of Shakespeare's Rhetoric: A Study of Nine Plays*. Swiss Studies in English, Volume 136. Tübingen: Narr.

Kilgarriff, A. (2001). 'English lexical sample task description', in *Proceedings of Senseval-2: 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*, Stroudsburg, PA: Association for Computational Linguistics, pp. 17–20.

Krippendorff, K. (1980). *Content Analysis: An Introduction to its Methodology*. Beverly Hills, CA: Sage.

Lagerwerf, L. (2002). 'Deliberate ambiguity in slogans: Recognition and appreciation'. *Document Design* 3 (3), pp. 245–260.

Lessard, G., Levison, M. & Venour, C. (2002). 'Cleverness versus funniness', in *Proceedings of the 20th Twente Workshop on Language Technology*, Enschede: Universiteit Twente, pp. 137–145.

Lippmann, L. G. & Dunn, M. L. (2000). 'Contextual connections within puns: Effects on perceived humor and memory'. *Journal of General Psychology* 127 (2), pp. 185–197.

Lucas, T. (2004). *Deciphering the Meaning of Puns in Learning English as a Second Language: A Study of Triadic Interaction*. Tallahassee, FL: Florida State University. PhD thesis.

Ludlow, P. J. (1996). 'Semantic Ambiguity and Underspecification' (review). *Computational Linguistics* 3 (23), pp. 476–482.

Mihalcea, R. & Chklovski, T. (2003). 'Open Mind Word Expert: Creating large annotated data collections with Web users' help', in *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC 2003)*, Stroudsburg, PA: Association for Computational Linguistics.

Mihalcea, R. & Strapparava, C. (2005). 'Making computers laugh: Investigations in automatic humor recognition', in *Proceedings of the 11th Human Language Technology Conference and the 10th Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 531–538.

Mihalcea, R. & Strapparava, C. (2006). 'Learning to laugh (automatically): Computational models for humor recognition'. *Computational Intelligence* 22 (2), pp. 126–142.

Miller, G. A., Leacock, C., Tengi, R. & Bunker, R. T. (1993). 'A semantic concordance', in *Proceedings of the 6th Human Language Technologies Conference (HLT 1993)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 303–308.

Miller, T., Biemann, C., Zesch, T. & Gurevych, I. (2012). 'Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation', in *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, Mumbai: COLING 2012 Organizing Committee, pp. 1781–1796.

Miller, T. & Gurevych, I. (2015). 'Automatic disambiguation of English puns', in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 719–729.

Monnot, M. (1981). *Selling America: Puns, Language and Advertising*. Washington, DC: University Press of America.

Monnot, M. (1982). 'Puns in advertising: Ambiguity as verbal aggression'. *Maledicta* 6, pp. 7–20.

Morkes, J., Kernal, H. K. & Nass, C. (1999). 'Effects of humor in task-oriented human–computer interaction and computer-mediated communication: A direct test of SRCT theory'. *Human–Computer Interaction* 14 (4), pp. 395–435.

Navigli, R. (2009). 'Word sense disambiguation: A survey'. *ACM Computing Surveys* 41, pp. 10:1–10:69.

Navigli, R. & Lapata, M. (2010). 'An experimental study of graph connectivity for unsupervised word sense disambiguation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (4), pp. 678–692.

Navigli, R., Litkowski, K. C. & Hargraves, O. (2007). 'SemEval-2007 Task 07: Coarse-grained English All-words Task', in *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 30–35.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L. & Dang, H. T. (2001). 'English tasks: All-words and verb lexical sample', in *Proceedings of Senseval-2: 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*, Stroudsburg, PA: Association for Computational Linguistics, pp. 21–24.

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.

Passonneau, R. J. (2006). 'Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation', in *Proceedings of the 5th International Conference on Language*

Resources and Evaluations (LREC 2006), Paris: European Language Resources Association, pp. 831–836.

Passonneau, R. J., Baker, C., Fellbaum, C. & Ide, N. (2012). 'The MASC word sense sentence corpus', in *Proceedings of the 8th International Conference on Language Resources and Evaluations (LREC 2012)*, Paris: European Language Resources Association, pp. 3025–3030.

Passonneau, R. J., Habash, N. & Rambow, O. (2006). 'Inter-annotator agreement on a multilingual semantic annotation task', in *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC 2006)*, Paris: European Language Resources Association, pp. 1951–1956.

Redfern, W. (1984). *Puns*. Oxford: Basil Blackwell.

Ritchie, G. D. (2004). *The Linguistic Analysis of Jokes*. London: Routledge.

Ritchie, G. D. (2005). 'Computational mechanisms for pun generation', in Wilcock, G., Jokinen, K., Mellish, C. & Reiter E. (eds.), *Proceedings of the 10th European Workshop on Natural Language Generation*, Stroudsburg, PA: Association for Computational Linguistics, pp. 8–10.

Rubinstein, F. (1984). *A Dictionary of Shakespeare's Sexual Puns and Their Significance*. London: Macmillan.

Schröter, T. (2005). *Shun the Pun, Rescue the Rhyme? The Dubbing and Subtitling of Language-Play in Film*. Karlstad: Karlstad University. PhD thesis.

Sharp, H. S. (1984). *Advertising Slogans of America*. Metuchen, NJ: Scarecrow Press.

Snyder, B. & Palmer, M. (2004). 'The English all-words task', in Mihalcea, R. & Edmonds, P. (eds.), *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 41–43.

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.

Valitutti, A., Strapparava, C. & Stock, O. (2008). 'Textual affect sensing for computational advertising', in *Proceedings of the AAAI Spring Symposium on Creative Intelligent Systems*, Menlo Park, CA: AAAI Press, pp. 117–122.

Venour, C. (1999). *The Computational Generation of a Class of Puns*. Kingston, ON: Queen's University. Master's thesis.

Wurth, L. (1895). *Das Wortspiel bei Shakespeare*. Vienna: Wilhelm Braumüller.

Yimam, S. M., Gurevych, I., de Castilho, R. E. & Biemann, C. (2013). 'WebAnno: A flexible, web-based and visually supported system for distributed annotations', in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, Stroudsburg, PA: Association for Computational Linguistics, pp. 1–6.

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.

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

Zwicky, A. M. & Zwicky, E. D. (1986). 'Imperfect puns, markedness, and phonological similarity: With fronds like these, who needs anemones?' *Folia Linguistica* 20 (3–4), pp. 493–503.