Automatic disambiguation of English puns

Tristan Miller and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de/

Abstract

Traditional approaches to 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. In this paper we describe how traditional, language-agnostic WSD approaches can be adapted to "disambiguate" puns, or rather to identify their double meanings. We evaluate several such approaches on a manually sense-annotated collection of English puns and observe performance exceeding that of some knowledge-based and supervised baselines.

1 Introduction

Word sense disambiguation, or WSD, is the task of identifying a word's meaning in context. No matter whether it is performed by a human or a machine, WSD usually rests on the assumption that there is a single unambiguous communicative intention underlying each word in the document.[1] However, there exists a class of language constructs known as *paronomasia* and *syllepsis*, or more generally as *puns*, in which homonymic (*i.e.*, coarse-grained) lexical-semantic ambiguity is a *deliberate* effect of the communication act. That is, the writer intends for a certain word or other lexical item to be interpreted as simultaneously carrying two or more separate meanings, or alternatively for it to be unclear which meaning is the intended one. There are a variety of motivations writers have for employing such constructions, and in turn for why such uses are worthy of scholarly investigation.

Perhaps surprisingly, this sort of intentional lexical ambiguity has attracted 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. A fundamental problem 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 instance, 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 (Valitutti et al., 2008). Recognizing instances of such lexical ambiguity and understanding their affective connotations would be of benefit to systems performing sentiment analysis on persuasive texts. Wordplay is also a perennial topic of scholarship in literary criticism and analysis. To give just one example, puns are one of the most intensively studied aspects of Shakespeare's rhetoric, and laborious manual counts have shown their frequency in certain

[1] 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 that 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 intentional lexical-semantic underspecification (Jurgens, 2014), it is 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.

of his plays to range from 17 to 85 instances per thousand lines (Keller, 2009). It is not hard to image how computer-assisted detection, classification, and analysis of puns could help scholars in the digital humanities. Finally, computational pun detection and understanding hold tremendous potential for machine-assisted translation. Some of the 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). These pose particular challenges for translators, who need not only to recognize and comprehend each instance of humour-provoking ambiguity, but also to select and implement an appropriate translation strategy.[2] 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.

In the present work, we discuss the adaptation of automatic word sense disambiguation techniques to intentionally ambiguous text and evaluate these adaptations in a controlled setting. We focus on humorous puns, as these are by far the most commonly encountered and more readily available in (and extractable from) existing text corpora.

The remainder of this paper is structured as follows: In the following section we give a brief introduction to puns, WSD, and related previous work on computational detection and comprehension of humour. In §3 we describe the data set produced for our experiments. In §§4 and 5 we describe how disambiguation algorithms, evaluation metrics, and baselines from traditional WSD can be adapted to the task of pun identification, and in §6 we report and discuss the performance of our adapted systems. Finally, we conclude in §7 with a review of our research contributions and an outline of our plans for future work.

2 Background

2.1 Puns

Punning is a form of wordplay where a word is used in such a way as to evoke several independent meanings simultaneously. Humorous and non-humorous puns have been the subject of extensive study in the humanities and social sciences, 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 (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, 1997).

Puns can be classified in various ways (Attardo, 1994), though from the point of view of our particular natural language processing application the most important distinction is between homographic and homophonic puns. A *homographic* pun exploits distinct meanings of the same written word, and a *homophonic* pun exploits distinct meanings of the same spoken word. Puns can be homographic, homophonic, both, or neither, as the following examples illustrate:

(1) A lumberjack's world revolves on its axes.

(2) She fell through the window but felt no pane.

(3) A political prisoner is one who stands behind her convictions.

(4) The sign at the nudist camp read, "Clothed until April."

In (1), the pun on *axes* is homographic but not homophonic, since the two meanings ("more than one axe" and "more than one axis") share the same spelling but have different pronunciations. In (2), the pun on *pane* ("sheet of glass") is homophonic but not homographic, since the word for the secondary meaning ("feeling of injury") is properly spelled *pain* but pronounced the same. The pun on *convictions* ("strongly held beliefs" and "findings of criminal guilt") in (3) is both homographic and homophonic. Finally, the pun on *clothed* in (4) is neither homographic nor homophonic, since the word for the secondary meaning, *closed*, differs in both spelling and pronunciation. Such puns 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 mul-

[2] The problem is compounded in audio-visual media such as films; often one or both of the pun's meanings appears in the visual channel, and thus cannot be freely substituted.

tiple parts of speech. We elaborate on the significance of these characteristics in the next section.

2.2 Word sense disambiguation

Word sense disambiguation (WSD) is the task of determining which sense of a polysemous term is the one intended when that term is used in a given communicative act. Besides the target term itself, a WSD system generally requires two inputs: the *context* (*i.e.*, the running text containing the target), and a *sense inventory* which specifies all possible senses of the target.

Approaches to WSD can be categorized according to the type of knowledge sources used to help discriminate senses. *Knowledge-based* approaches restrict themselves to using pre-existing lexical-semantic resources (LSRs), or such additional information as can be automatically extracted or mined from raw text corpora. *Supervised* approaches, on the other hand, use manually sense-annotated corpora as training data for a machine learning system, or as seed data for a bootstrapping process. Supervised WSD systems generally outperform their knowledge-based counterparts, though this comes at the considerable expense of having human annotators manually disambiguate hundreds or thousands of example sentences. Moreover, supervised approaches tend to be such that they can disambiguate only those words for which they have seen sufficient training examples to cover all senses. That is, most of them cannot disambiguate words which do not occur in the training data, nor can they select the correct sense of a known word if that sense was never observed in the training data.

Regardless of the approach, all WSD systems work by extracting contextual information for the target word and comparing it against the sense information stored for that word. A seminal knowledge-based example is the Lesk algorithm (Lesk, 1986) which disambiguates a pair of target terms in context by comparing their respective dictionary definitions and selecting the two with the greatest number of words in common. Though simple, the Lesk algorithm performs surprisingly well, and has frequently served as the basis of more sophisticated approaches. In recent years, Lesk variants in which the contexts and definitions are supplemented with entries from a distributional thesaurus (Lin, 1998) have achieved state-of-the-art performance for knowledge-based systems on standard data sets (Miller et al., 2012; Basile et al., 2014).

In traditional word sense disambiguation, the part of speech and lemma of the target word are usually known *a priori*, or can be determined with high accuracy using off-the-shelf natural language processing tools. The pool of candidate senses can therefore be restricted to those whose lexicalizations exactly match the target lemma and part of speech. No such help is available for puns, at least not in the general case. Take the following two examples:

(5) Tom moped.

(6) "I want a scooter," Tom moped.

In the first of these sentences, the word *moped* is unambiguously a verb with the lemma *mope*, and would be correctly recognized as such by any automatic lemmatizer and part-of-speech tagger. The *moped* of the second example is a pun, one of whose meanings is the same inflected form of the verb *mope* ("to sulk") and the other of which is the noun *moped* ("motorized scooter"). For such cases an automated pun identifier would therefore need to account for all possible lemmas for all possible parts of speech of the target word. The situation becomes even more onerous for heterographic and imperfect puns, which may require the use of pronunciation dictionaries, and application of phonological theories of punning, in order to recover the lemmas (Hempelmann, 2003).

As our research interests are in lexical semantics rather than phonology, we focus on puns which are homographic and monolexemic. This allows us to investigate the problem of pun identification in as controlled a setting as possible.

2.3 Previous work

2.3.1 Computational humour

There is some previous research on computational detection and comprehension of humour, though by and large it is not concerned specifically with puns; those studies which do analyze puns tend to have a phonological or syntactic rather than semantic bent. In this subsection we briefly review some prior work which is relevant to ours.

Yokogawa (2002) describes a system for detecting the presence of puns in Japanese text. However, this work is concerned only with puns which are both imperfect and ungrammatical, relying on syntactic cues rather than the lexical-semantic information we propose to use. Taylor and Mazlack (2004)

describe an n-gram–based approach for recognizing when imperfect puns are used for humorous effect in a certain 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 perfect puns in running text. Mihalcea and Strapparava (2005) treat humour recognition as a classification task, employing various machine learning techniques on humour-specific stylistic features such as alliteration and antonymy. Of particular interest is their follow-up analysis (Mihalcea and Strapparava, 2006), where they specifically point to their system's failure to resolve lexical-semantic ambiguity as a stumbling block to better accuracy, and speculate that deeper semantic analysis of the text, such as via word sense disambiguation or domain disambiguation, could aid in the detection of humorous incongruity and opposition.

The previous work which is perhaps most relevant to ours is that of Mihalcea et al. (2010). They build a data set consisting of 150 joke set-ups, each of which is followed by four possible "punchlines", only one of which is actually humorous (but not necessarily due to a pun). They then compare the set-ups against the punchlines using various models of incongruity detection, including many exploiting knowledge-based semantic relatedness such as Lesk. The Lesk model had an accuracy of 56%, which is lower than that of a naïve polysemy model which simply selects the punchline with the highest mean polysemy (66%) and even of a random-choice baseline (62%). However, it should be stressed here that the Lesk model did not directly account for the possibility that any given word might be ambiguous. Rather, for every word in the setup, the Lesk measure was used to select a word in the punchline such that the lexical overlap between each *one* of their possible definitions was maximized. The overlap scores for all word pairs were then averaged, and the punchline with the lowest average score selected as the most humorous.

2.3.2 Corpora

There are a number of English-language corpora of intentional lexical ambiguity 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 Swifty" puns 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 data set 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 specialized dictionary. 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. Two studies on cognitive strategies used by second language learners (Kaplan and Lucas, 2001; Lucas, 2004) used a data set of 58 jokes compiled from newspaper comics, 32 of which rely on lexical ambiguity. Bucaria (2004) conducts a linguistic analysis of a set of 135 humorous newspaper headlines, about half of which exploit lexical ambiguity.

Such data sets—particularly the larger ones—provided us good evidence that intentionally lexical ambiguous exemplars exist in sufficient numbers to make a rigorous evaluation of our task feasible. Unfortunately, none of the above-mentioned corpora have been published in full, and moreover many of them contain (sometimes exclusively) the sort of imperfect or otherwise heterographic puns which we mean to exclude from consideration. This has motivated us to produce our own corpus of puns, the construction and analysis of which is described in the following section.

3 Data set

As in traditional WSD, a prerequisite for our research is a corpus of examples, where one or more human annotators have already identified the ambiguous words and marked up their various meanings with reference to a given sense inventory. Such a corpus is sufficient for evaluating what we term *pun identification* or *pun disambiguation*—that is, identifying the senses of a term known *a priori* to be a pun.

3.1 Construction

Though several prior studies have produced corpora of puns, none of them are systematically sense-annotated. We therefore compiled our own corpus by pooling together some of the aforementioned

corpora, the user-submitted puns from the Pun of the Day website,[3] and private collections provided to us by some professional humorists. This raw collection of 7750 one-liners was then filtered by trained human annotators to those instances 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. (This criterion simplifies the task detecting the presence and location of puns in a text, a classification task which we intend to investigate in future work.)

One content word per pun: The lexical unit that forms 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. This criterion is important because, in our observations, it is often only one word which carries ambiguity in puns on compounds and multi-word expressions. Accepting lexical units containing more than one content word would have required our annotators to laboriously partition the pun into (possibly overlapping) sense-bearing units and to assign sense sets to each of them, inflating the complexity of the annotation task to unacceptable levels.

Two meanings per pun: The pun must have exactly two distinct meanings. Though many sources state that puns have only two senses (Redfern, 1984; Attardo, 1994), our annotators identified a handful of corpus 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.

Weak homography: The lexical units corresponding to the two distinct meanings must be spelled exactly the same way, except 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 recognized by our human annotators.

The filtering reduced the number of instances to 1652, whose puns two human judges annotated with sense keys from WordNet 3.1 (Fellbaum, 1998). Using an online annotation tool specially constructed for this study, the annotators applied two sets of sense keys to each instance, one for each of the two meanings of the pun. For cases where the distinction between WordNet's fine-grained senses was irrelevant, the annotators had the option of labelling the meaning with more than one sense key. Annotators also had the option of marking a meaning as unassignable if WordNet had no corresponding sense key. Further details of our annotation tool and its use can be found in Miller and Turković (2015).

3.2 Analysis

Our judges agreed on which word was the pun in 1634 out of 1652 cases, a raw agreement of 98.91%. For the agreed cases, we used DKPro Agreement (Meyer et al., 2014) to compute Krippendorff's α (Krippendorff, 1980) for the sense annotations. This is a chance-correcting metric of inter-annotator agreement 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: $d_{M'}(\{A_1,A_2\},\{B_1,B_2\}) = \frac{1}{2}\min(d_M(A_1,B_1) + d_M(A_2,B_2), d_M(A_1,B_2) + d_M(A_2,B_1))$. 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 and Klapaftis, 2013).

Where possible, we resolved sense annotation disagreements automatically by taking the intersection of corresponding sense sets. Where the annotators' sense sets were disjoint or contradictory (including the cases where the annotators disagreed on the pun word), we had a human adjudicator attempt to resolve the disagreement in favour of one annotator or the other. This left us with 1607 instances,[4] of which we retained only the 1298 that had successful (*i.e.*, not marked as unassignable)

[3] `http://www.punoftheday.com/`

[4] Pending clearance of the distribution rights, we will make some or all of our annotated data set available on our website at `https://www.ukp.tu-darmstadt.de/data/`.

annotations for the present study. The contexts in this data set range in length from 3 to 44 words, with an average length of 11.9. The 2596 meanings carry sense key annotations corresponding to anywhere from one to seven WordNet synsets, with an average of 1.08. As expected, then, WordNet's sense granularity proved to be somewhat finer than necessary to characterize the meanings in the data set, though only marginally so.

Of the 2596 individual meanings, 1303 (50.2%) were annotated with noun senses only, 877 (33.8%) with verb senses only, 340 (13.1%) with adjective senses only, and 41 (1.6%) with adverb senses only. Only 35 individual meanings (1.3%) carry sense annotations corresponding to multiple parts of speech. However, for 297 (22.9%) of our puns, the two meanings had different parts of speech. Similarly, sense annotations for each individual meaning correspond to anywhere from one to four different lemmas, with a mean of 1.25. These observations confirm the concerns we raised in §2.2 that pun disambiguators, unlike traditional WSD systems, cannot always rely on the output of a lemmatizer or part-of-speech tagger to narrow down the list of sense candidates.

4 Pun disambiguation

It has long been observed that gloss overlap–based WSD systems, such as those based on the Lesk algorithm, fail to distinguish between candidate senses when their definitions have a similar overlap with the target word's context. In some cases this is because the overlap is negligible or nonexistent; this is known as the *lexical gap* problem, and various solutions to it are discussed in (*inter alia*) Miller et al. (2012). In other cases, the indecision arises because the definitions provided by the sense inventory are too fine-grained; this problem has been addressed, with varying degrees of success, through sense clustering or coarsening techniques (a short but reasonably comprehensive survey of which appears in Matuschek et al. (2014)). A third condition under which senses cannot be discriminated is when the target word is used in an underspecified or intentionally ambiguous manner. We hold that for this third 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 of senses), we can allow disambiguation systems to sense-annotate intentionally ambiguous constructions such as puns.

Many approaches to WSD, including Lesk-like algorithms, 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. Accordingly we applied this modification to the following knowledge-based WSD algorithms:

Simplified Lesk (Kilgarriff and Rosenzweig, 2000) disambiguates a target word by examining the definitions[5] for each of its candidate senses and selecting the single sense—or in our case, the two senses—which have the greatest number of words in common with the context. As we previously demonstrated that puns often transcend part of speech, our pool of candidate senses is constructed as follows: we apply a morphological analyzer to recover all possible lemmas of the target word without respect to part of speech, and for each lemma we add all its senses to the pool.

Simplified extended Lesk (Ponzetto and Navigli, 2010) is similar to simplified Lesk, except that the definition for each sense is concatenated with those of neighbouring senses in WordNet's semantic network.

Simplified lexically expanded Lesk (Miller et al., 2012) is also based on simplified Lesk, with the extension that every word in the context and sense definitions is expanded with up to 100 entries from a large distributional thesaurus.

The above algorithms fail to make a sense assignment when more than two senses are tied for the highest lexical overlap, or when there is a single highest-scoring sense but multiple senses are tied for the second-highest overlap. We therefore devised two pun-specific tie-breaking strategies. The first is motivated by the informal observation that, though the two meanings of a pun may have different parts of speech, at least one of the parts

[5] In our implementation, the sense definitions are formed by concatenating the synonyms, gloss, and example sentences provided by WordNet.

of speech is grammatical in the context of the sentence, and so would probably be the one assigned by a stochastic or rule-based POS tagger. Our "POS" tie-breaker therefore preferentially selects the best sense, or pair of senses, whose POS matches the one applied to the target by the Stanford POS tagger (Toutanova et al., 2003). For our second tie-breaking strategy, we posit that since humour derives from the resolution of semantic incongruity (Raskin, 1985; Attardo, 1994), puns are more likely to exploit coarse-grained homonymy than than fine-grained systematic polysemy. Thus, following Matuschek et al. (2014), we induced a clustering of WordNet senses by aligning WordNet to the more coarse-grained OmegaWiki LSR.[6] Our "cluster" fallback works the same as the "POS" one, with the addition that any remaining ties among senses with the second-highest overlap are resolved by preferentially selecting those which are not in the same induced cluster as, and which in WordNet's semantic network are at least three edges distant from, the sense with the highest overlap.

5 Evaluation

5.1 Scoring

In traditional word sense disambiguation, *in vitro* evaluations are conducted by comparing the senses assigned by the disambiguation system to the gold-standard senses assigned by the human annotators. 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 coverage (the number of targets for which a sense assignment was attempted), precision (the sum of scores divided by the number of attempted targets), recall (the sum of scores divided by the total number of targets in the data set), and F_1 (the harmonic mean of precision and recall) (Palmer et al., 2006).

The traditional approach to scoring individual targets is not usable as-is for pun disambiguation, because each pun carries two disjoint but equally valid sets of sense annotations. Instead, since our systems assign exactly one sense to each of the pun's two sense sets, we 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. (As with traditional WSD scoring, various approaches could be used to assign credit for partially correct assignments, though we leave exploration of these to future work.)

[6] `http://www.omegawiki.org/`

5.2 Baselines

System performance in WSD is normally interpreted with reference to one or more baselines. To our knowledge, ours is the very first study of automatic pun disambiguation on any scale, so at this point there are no previous systems against which to compare our results. However, traditional WSD systems are often compared with two naïve baselines (Gale et al., 1992) which can be adapted for our purposes.

The first of these naïve baselines is to randomly select from among the candidate senses. In traditional WSD, the score for a random disambiguator which selects a single sense for a given target t is the number of gold-standard senses divided by the number of candidate senses: $\text{score}(t) = g(t) \div \delta(t)$. In our pun disambiguation task, however, a random disambiguator must select *two* senses—one for each of the sense sets $g_1(t)$ and $g_2(t)$—and these senses must be distinct. There are $\binom{\delta(t)}{2}$ possible ways of selecting two unique senses, so the random score for any given instance is $\text{score}(t) = g_1(t) \cdot g_2(t) \div \binom{\delta(t)}{2}$.

The second naïve baseline for WSD, known as *most frequent sense* (MFS), is a supervised baseline, meaning that it depends on a manually sense-annotated background corpus. As its name suggests, it involves always selecting from the candidates that sense which has the highest frequency in the corpus. As with our test algorithms, we adapt this technique to pun disambiguation by having it select the two most frequent senses (according to WordNet's built-in sense frequency counts). In traditional WSD, MFS baselines are notoriously difficult to beat, even for supervised disambiguation systems, and since they rely on expensive sense-tagged data they are not normally considered a benchmark for the performance of knowledge-based disambiguators.

system	C	P	R	F_1
SL	35.52	19.74	7.01	10.35
SEL	42.45	19.96	8.47	11.90
SLEL	**98.69**	13.43	13.25	13.34
SEL+POS	59.94	**21.21**	12.71	15.90
SEL+cluster	68.10	20.70	**14.10**	**16.77**
random	100.00	9.31	9.31	9.31
MFS	100.00	13.25	13.25	13.25

Table 1: Coverage, precision, recall, and F_1 for various pun diasmbiguation algorithms.

6 Results

Using the freely available DKPro WSD framework (Miller et al., 2013), we implemented our pun disambiguation algorithms, ran them on our full data set, and compared their annotations against those of our manually produced gold standard. Table 1 shows the coverage, precision, recall, and F_1 for simplified Lesk (SL), simplified extended Lesk (SEL), simplified lexically expanded Lesk (SLEL), and the random and most frequent sense baselines; for SEL we also report results for each of our pun-specific tie-breaking strategies. All metrics are reported as percentages, and the highest score for each metric (excluding baseline coverage, which is always 100%) is highlighted in boldface.

Accuracy for the random baseline annotator was about 9%; for the MFS baseline it was just over 13%. These figures are considerably lower than what is typically seen with traditional WSD corpora, where random baselines achieve accuracies of 30 to 60%, and MFS baselines 65 to 80% (Palmer et al., 2001; Snyder and Palmer, 2004; Navigli et al., 2007). Our baselines' low figures are the result of them having to consider senses from every possible lemmatization and part of speech of the target, and underscore the difficulty of our task.

The simplest knowledge-based algorithm we tested, simplified Lesk, was over twice as accurate as the random baseline in terms of precision (19.74%), but predictably had very low coverage (35.52%), leading in turn to very low recall (7.01%). Manual examination of the unassigned instances confirmed that failure was usually due to the lack of any lexical overlap whatsoever between the context and definitions. The use of a tie-breaking strategy would not help much here, though some way of bridging the lexical gap would. This is, in fact, the strategy employed by the extended and lexically expanded variants of simplified Lesk, and we observed that both were successful to some degree. Simplified lexically expanded Lesk almost completely closed the lexical gap, with nearly complete coverage (98.69%), though this came at the expense of a large drop in precision (to 13.43%). Given the near-total coverage, use of a tie-breaking strategy here would have no appreciable effect on the accuracy.

Simplified extended Lesk, on the other hand, saw significant increases in coverage, precision, and recall (to 42.45%, 19.96%, and 8.47%, respectively). Its recall is statistically indistinguishable[7] from the random baseline, though spot-checks of its unassigned instances show that the problem is very frequently not the lexical gap but rather multiple senses tied for the greatest overlap with the context. We therefore tested our two pun-specific backoff strategies to break this system's ties. Using the "POS" strategy increased coverage by 41%, relatively speaking, and gave us our highest observed precision of 21.21%. Our "cluster" strategy effected a relative increase in coverage of over 60%, and gave us the best recall (14.10%). This strategy also had the best tradeoff between precision and recall, with an F_1 of 16.77%.

Significance testing shows the recall scores for SLEL, SEL+POS, and SEL+cluster to be significantly better than the random baseline, and statistically indistinguishable from that of MFS. This is excellent news, especially in light of the fact that supervised approaches (even baselines like MFS) usually outperform their knowledge-based counterparts. Though the three knowledge-based systems are not statistically distinguishable from each other in terms of recall, they do show a statistically significant improvement over SL and SEL, and the two implementing pun-specific tie-breaking strategies were markedly more accurate than SLEL for those targets where they attempted an assignment. These two systems would therefore be preferable for applications where precision is more important than recall.

We also examined the results of our generally best-performing system, SEL+cluster, to see whether there was any relationship with the targets' part of speech. We filtered the results according to whether both gold-standard meanings of the pun contain senses for nouns only, verbs only, adjec-

[7] All significance statements in this section are based on McNemar's test at a confidence level of 5%.

POS	C	P	R	R_{rand}
noun	66.60	20.89	13.91	10.44
verb	65.61	14.54	9.54	5.12
adj.	68.87	39.73	27.36	16.84
adv.	100.00	75.00	75.00	46.67
pure	66.77	21.44	14.31	9.56
mult.	72.58	18.43	13.38	12.18

Table 2: Coverage, precision, and recall for SEL+cluster, and random baseline recall, according to part of speech.

tives only, or adverbs only; these amounted to 539, 346, 106, and 8 instances, respectively. These results are shown in Table 2. Also shown there is a row which aggregates the 999 targets with "pure" POS, and another for the remaining 608 instances ("mult."), where one or both of the two meanings contain senses for multiple parts of speech, or where the two meanings have different parts of speech. The last column of each row shows the recall of the random baseline for comparison.

Accuracy was lowest on the verbs, which had the highest candidate polysemy (21.6) and are known to be particularly difficult to disambiguate even in traditional WSD. Still, as with all the other single parts of speech, performance of SEL+cluster exceeded the random baseline. While recall was lower on targets with mixed POS than those with pure POS, coverage was significantly higher. Normally such a disparity could be attributed to a difference in polysemy: Lesk-like systems are more likely to attempt a sense assignment for highly polysemous targets, since there is a greater likelihood of one of the candidate definitions matching the context, though the probability of the assignment being correct is reduced. In this case, however, the multi-POS targets actually had lower average polysemy than the single-POS ones (13.2 *vs.* 15.8).

7 Conclusion

In this paper we have introduced the novel task of pun disambiguation and have proposed and evaluated several computational approaches for it. The major contributions of this work are as follows: First, we have produced a new data set consisting of manually sense-annotated homographic puns. The data set is large enough, and the manual annotations reliable enough, for a principled evaluation of automatic pun disambiguation systems. Second, we have shown how evaluation metrics, baselines, and disambiguation algorithms from traditional WSD can be adapted to the task of pun disambiguation, and we have tested these adaptations in a controlled experiment. The results show pun disambiguation to be a particularly challenging task for NLP, with baseline results far below what is commonly seen in traditional WSD. We showed that knowledge-based disambiguation algorithms naïvely adapted from traditional WSD perform poorly, but that extending them with strategies that rely on pun-specific features brings about dramatic improvements in accuracy: their recall becomes comparable to that of a supervised baseline, and their precision greatly exceeds it.

There are a number of avenues we intend to explore in future work. First, we would like to try adapting and evaluating some additional WSD algorithms for use with puns. Though our data set is probably too small to use with machine learning–based approaches, we are particularly interested in testing knowledge-based disambiguators which rely on measures of graph connectivity rather than gloss overlaps. Second, we would like to investigate alternative tie-breaking strategies, such as the domain similarity measures used by Mihalcea et al. (2010). Finally, whereas in this paper we have treated only the task of sense disambiguation for the case where a word is known *a priori* to be a pun, we are interested in exploring the requisite problem of *pun detection*, where the object is to determine whether or not a given context contains a pun, and more precisely whether any given word in a context is a pun.

Acknowledgments

The work described in this paper is 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, Mladen Turković, and Andreas Zimpfer for helping us build our data set.

References

Salvatore Attardo. 1994. *Linguistic Theories of Humor*. Mouton de Gruyter.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disam-

biguation algorithm through a distributional semantic model. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1591–1600.

Tristan A. Bekinschtein, Matthew H. Davis, Jennifer M. Rodd, and Adrian M. Owen. 2011. Why clowns taste funny: The relationship between humor and semantic ambiguity. *The Journal of Neuroscience*, 31(26):9665–9671, June.

Nancy D. Bell, Scott Crossley, and Christian F. Hempelmann. 2011. Wordplay in church marquees. *Humor: International Journal of Humor Research*, 24(2):187–202, April.

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

Paul Buitelaar. 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, pages 14–19.

Jonathan D. Culler, editor. 1988. *On Puns: The Foundation of Letters*. Basil Blackwell, Oxford.

Dirk Delabastita, editor. 1997. *Traductio: Essays on Punning and Translation*. St. Jerome, Manchester.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

William Gale, Kenneth Ward Church, and David Yarowsky. 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)*, pages 249–256.

Christian F. Hempelmann. 2003. *Paronomasic Puns: Target Recoverability Towards Automatic Generation*. Ph.D. thesis, Purdue University.

Bryan Anthony Hong and Ethel Ong. 2009. Automatically extracting word relationships as templates for pun generation. In *Proceedings of the 1st Workshop on Computational Approaches to Linguistic Creativity (CALC 2009)*, pages 24–31, June.

David Jurgens and Ioannis Klapaftis. 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)*, pages 290–299, June.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 3006–3012, May.

Nora Kaplan and Teresa Lucas. 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):245–258.

Stefan Daniel Keller. 2009. *The Development of Shakespeare's Rhetoric: A Study of Nine Plays*, volume 136 of *Swiss Studies in English*. Narr, Tübingen.

Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34:15–48.

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

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

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of the 5th Annual International Conference of Systems Documentation (SIGDOC 1986)*, pages 24–26.

Greg Lessard, Michael Levison, and Chris Venour. 2002. Cleverness versus funniness. In *Proceedings of the 20th Twente Workshop on Language Technology*, pages 137–145.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL 1998) and the 17th International Conference on Computational Linguistics (COLING 1998)*, volume 2, pages 768–774.

Teresa Lucas. 2004. *Deciphering the Meaning of Puns in Learning English as a Second Language: A Study of Triadic Interaction*. Ph.D. thesis, Florida State University.

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

Michael Matuschek, Tristan Miller, and Iryna Gurevych. 2014. A language-independent sense clustering approach for enhanced WSD. In *Proceedings of the 12th Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2014)*, pages 11–21, October.

Christian M. Meyer, Margot Mieskes, Christian Stab, and Iryna Gurevych. 2014. DKPro Agreement: An open-source Java library for measuring inter-rater agreement. In *Proceedings of the 25th International Conference on Computational Linguistics (System Demonstrations) (COLING 2014)*, pages 105–109, August.

Rada Mihalcea and Carlo Strapparava. 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)*, pages 531–538, October.

Rada Mihalcea and Carlo Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence*, 22(2):126–142.

Rada Mihalcea, Carlo Strapparava, and Stephen Pulman. 2010. Computational models for incongruity detection in humour. In *Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2010)*, volume 6008 of *Lecture Notes in Computer Science*, pages 364–374. Springer, March.

Tristan Miller and Mladen Turković. 2015. Towards the automatic detection and identification of English puns. *European Journal of Humour Research*. To appear.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 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)*, pages 1781–1796, December.

Tristan Miller, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych. 2013. DKPro WSD: A generalized UIMA-based framework for word sense disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 37–42, August.

Michel Monnot. 1982. Puns in advertising: Ambiguity as verbal aggression. *Maledicta*, 6:7–20.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. SemEval-2007 Task 07: Coarse-grained English All-words Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, June.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of Senseval-2: 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24, July.

Martha Palmer, Hwee Tou Ng, and Hoa Trang Dang. 2006. Evaluation of WSD systems. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech, and Language Technology*. Springer.

Rebecca J. Passonneau, Nizar Habash, and Owen Rambow. 2006. Inter-annotator agreement on a multilingual semantic annotation task. In *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC 2006)*, pages 1951–1956.

Rebecca J. Passonneau. 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)*, pages 831–836.

Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1522–1531.

Vitor Raskin. 1985. *Semantic Mechanisms of Humor*. D. Reidel, Dordrecht, the Netherlands.

Walter Redfern. 1984. *Puns*. Basil Blackwell, Oxford.

Thorsten Schröter. 2005. *Shun the Pun, Rescue the Rhyme? The Dubbing and Subtitling of Language-Play in Film*. Ph.D. thesis, Karlstad University.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 41–43, July.

Julia M. Taylor and Lawrence J. Mazlack. 2004. Computationally recognizing wordplay in jokes. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society (CogSci 2004)*, pages 1315–1320, August.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 3rd Conference of the North American Chapter of the Association for Computational Linguistics and the 9th Human Language Technologies Conference (HLT-NAACL 2003)*, pages 252–259.

Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. 2008. Textual affect sensing for computational advertising. In *Proceedings of the AAAI Spring Symposium on Creative Intelligent Systems*, pages 117–122, March.

Chris Venour. 1999. The computational generation of a class of puns. Master's thesis, Queen's University, Kingston, ON.

Toshihiko Yokogawa. 2002. Japanese pun analyzer using articulation similarities. In *Proceedings of the 11th IEEE International Conference on Fuzzy Systems (FUZZ 2002)*, volume 2, pages 1114–1119, May.

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