Incremental dependency parsing and disfluency detection in spoken learner English

Russell Moore, Andrew Caines, Calbert Graham, and Paula Buttery

Automated Language Teaching & Assessment Institute, Department of Theoretical & Applied Linguistics, University of Cambridge, Cambridge, U.K.
{rjm49,apc38,crg29,pjb48}@cam.ac.uk

Abstract. This paper investigates the suitability of state-of-the-art natural language processing (NLP) tools for parsing the spoken language of second language learners of English. The task of parsing spoken learner-language is important to the domains of automated language assessment (ALA) and computer-assisted language learning (CALL). Due to the non-canonical nature of spoken language (containing filled pauses, non-standard grammatical variations, hesitations and other disfluencies) and compounded by a lack of available training data, spoken language parsing has been a challenge for standard NLP tools. Recently the Redshift parser (Honnibal *et al*. In: *Proceedings of CoNLL* (2013)) has been shown to be successful in identifying grammatical relations and certain disfluencies in native speaker spoken language, returning unlabelled dependency accuracy of 90.5% and a disfluency F-measure of 84.1% (Honnibal & Johnson: *TACL* 2, 131-142 (2014)). We investigate how this parser handles spoken data from learners of English at various proficiency levels. Firstly, we find that Redshift's parsing accuracy on non-native speech data is comparable to Honnibal & Johnson's results, with 91.1% of dependency relations correctly identified. However, disfluency detection is markedly down, with an F-measure of just 47.8%. We attempt to explain why this should be, and investigate the effect of proficiency level on parsing accuracy. We relate our findings to the use of NLP technology for CALL and ALA applications.

Keywords: spoken language, learner English, learner proficiency, disfluency detection, dependency parsing

1 Introduction

Most natural language processing (NLP) experiments are carried out with tools trained on (mainly written) native speaker data. Two corpora in particular, Brown and the Wall Street Journal, have been widely used for the evaluation of NLP technology [24], with the WSJ being near-ubiquitous in parser testing [5, 19].

There have been various efforts to extend the range of domains on which NLP tools are trained and evaluated, including, for example, biomedical literature [21, 28], Twitter posts [12, 17], and spoken language [8, 16]. Each new domain presents its own challenges in terms of NLP: spoken language, for instance, differs from canonical written data in multiple ways [2, 3, 9]. Disfluencies are an especially characteristic feature of

speech, and have been the object of interest for several NLP studies, as their presence is disruptive to parsing, with knock-on effects for any applications that follow.

The convention set by Shriberg [29], and applied to a portion of the Switchboard Corpus (SWB; [14]), is to differentiate between filled pauses (FP), the 'reparandum' (RM), 'interregnum' (IM) and repair (RP) in disfluent sections of text. This annotation scheme is exemplified in Figure 1.

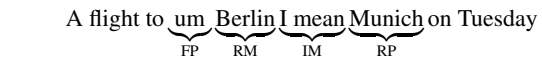

Fig. 1. Disfluency example, annotated in the Switchboard Corpus style.

The goal then of automated disfluency detection is to identify FPs, RMs and IMs – *i.e.* here 'um Berlin I mean' – so that the resultant string is understood to actually be 'A flight to Munich on Tuesday'. Several recent papers have reported various approaches achieving upwards of four-in-five accuracy in this task [16, 26, 27]. For instance, Honnibal & Johnson (H&J) report how well an incremental dependency parsing model, 'Redshift' [15], was able to identify dependency relations and speech repairs in hand-annotated sections of SWB, with an unlabelled attachment score (UAS)[1] of 90.5% and 84% F-measure for disfluency detection [16].

All previous efforts in disfluency detection have targeted the transcribed speech of native speakers of English. We apply the Redshift parser to *non*-native speaker (or, 'learner') English, transcribed from business English oral examinations. The texts come from learners of different proficiency levels: we asked firstly whether Redshift would be able to handle non-native speaker data as accurately as that of native speakers, and secondly whether speaker proficiency would affect parsing accuracies.

We hypothesise that Redshift should be able to parse the language of higher proficiency learners more accurately, as this is presumed to more closely approximate the language of native speakers, on which Redshift is trained. Such an outcome would be in line with previous work showing that the RASP System [5] was better able to parse higher proficiency texts [8].

We indeed found that Redshift could analyse our non-native speaker data remarkably well, with a UAS of 91.1%. However, disfluency detection was less successful, with an F-measure of just 47.8%. We propose that this is due to learner disfluencies being more extended and less orderly than those of native speakers. As for learner proficiency, there is a general upward trend in UAS and disfluency detection as the level moves from CEFR[2] B1 'intermediate', to B2 'upper intermediate', to C1 'advanced' [11]. This finding has implications both as a diagnostic tool for learner proficiency, in

[1] The percentage of tokens with a correctly-identified head word (*labelled* attachment is another commonly-reported metric (LAS); this is the percentage of tokens with correctly-identified head word *and* dependency relation) [20].

[2] The 'Common European Framework of Reference for Languages': a schema for grading an individual learner's language level. For further information go to `http://www.coe.int/lang-CEFR`

the context of automated language assessment (ALA), as well as the automated provision of feedback to learners on ways to improve, of the kind required by computer-assisted language learning (CALL) systems.

2 Transition-based dependency parsing

H&J's Redshift parser [16] is a transition-based dependency parser modelled on Zhang & Clark's design with a structured average perceptron for training and beam search for decoding [31]. Syntactic structure is predicted incrementally, based on a series of classification decisions as to which parsing action to take with regard to tokens on the 'stack' and in the 'buffer', two disjoint sets of word indices to the right and left of the current token.

Redshift adopts the four actions defined in Nivre's arc-eager transition system [25] – SHIFT, LEFT-ARC, RIGHT-ARC, REDUCE – and adds a novel non-monotonic EDIT transition to repair disfluencies during parsing [15]. SHIFT moves the first item of the buffer onto the stack. RIGHT-ARC does the same, adding an arc so that items 1 and 2 on the stack are connected. LEFT-ARC and REDUCE both pop the stack, with the former first adding an arc from word 1 in the buffer to word 1 on the stack. Like [1], Redshift posits a dummy ROOT token to govern the head-word of each utterance; with the ROOT token at the top of the buffer and an empty stack, the parsing of this utterance ends.

The novel EDIT transition marks the token on top of the stack as disfluent along with any rightward descendents to the start of the buffer. The stack is then popped and any dependencies to or from the deleted tokens are erased. The transition is 'non-monotonic': previously-created dependencies may be deleted and previously-popped tokens are returned to the stack (for further detail and worked examples see [15, 16]).

3 Experiments

3.1 Datasets

Switchboard Corpus The Switchboard Corpus (SWB) is a collection of transcribed two-way telephone conversations among a network of hundreds of unacquainted English speaking volunteers from across the U.S.A. [14]. The calls were computer-operated, such that no two speakers spoke together more than once, and so that no single speaker was given the same topic prompt (of which there were about 70) more than once [13]. The entire corpus contains 2320 conversations of approximately 5 minutes each, totalling about 3 million words of transcribed text.

Our focus is on a subset of SWB that was hand-annotated with both syntactic bracketing and disfluency labels as part of the Penn Treebank project [30]. The project, now discontinued, produced approximately 1.5 million words from the SWB transcripts annotated for disfluencies. But as hand-labelled syntactic bracketing is more expensive to produce, only 0.6 million tokens from the disfluency layer have corresponding syntactic annotations. Following H&J [16], we require the smaller dataset with *both* syntactic and disfluency annotations (SWB-SYN-DISF), using this as training data, which we

note gives us less than half the training data used in other state-of-the-art disfluency detection systems [26, 27].

We obtained SWB-SYN-DISF in CoNLL-X treebank format [6]: the same dataset that featured in [16] and which we use here to train Redshift[3]. This dataset was pre-processed in the following ways: removal of filled pauses such as 'uh' and 'um', one-token sentences and partial words, all text to lower-case, punctuation stripped, and 'you know' 'i mean' bigrams merged to single-token 'you_know' and 'i_mean'.

BULATS Corpus Our non-native speaker data is provided by Cambridge English, University of Cambridge[4]. The corpus is a collection of transcribed recordings from their Business Language Testing Service (BULATS) speaking tests. We produced a gold-standard BULATS treebank of 5667 tokens, annotated for the same features and in the same format as SWB-SYN-DISF.

The dataset features speakers from Pakistan, India and Brazil, with Gujarati, Urdu, Sindhi, Hindi, Portuguese and Panjanbi as their first languages. The BULATS corpus was pre-processed in the same way as SWB-SYN-DISF, described above, with the removal of filled pauses, *etc*. It contains transcripts from 16 different learners and is divided into approximately 1900 tokens from each of three CEFR levels: B1, B2 and C1. We refer to these data subsets as BULATS:B1, BULATS:B2, BULATS:C1.

3.2 Procedure

System Setup Redshift is an open-source parser so we were able to obtain the same code-base that was used in [16]. We also trained the parser in a similar manner[5]: the training data comprises the 600k token SWB-SYN-DISF dataset. The feature set, *disfl*, is an extended version of 73 templates from [32] mostly pertaining to local context as represented by twelve context tokens, plus extra features to detect 'rough copy' edits [18] and contiguous bursts of disfluency. The EDIT transition was enabled, allowing the parser to erase disfluent tokens from the utterance. Random seed training was disabled, since we only trained the parser once (whereas H&J averaged over 20 random seeds).

We used 15 iterations to train the perceptron, and set the beam width to 32 so that the 32 best-scoring transition candidates were kept in the beam with each iteration. Redshift utilises its own part-of-speech tagger – also guided by averaged perceptron and beam search decoding – which was set to be trained in unison with the parser.

Task 1: Dependency relations As noted in [16], Redshift makes use of the Stanford Basic Dependencies scheme (SD), which makes strictly projective representations of grammatical relations. The dependencies indicate the binary relations between tokens

[3] Our profound thanks to Matthew Honnibal for sharing this data, and for his help in setting up Redshift.

[4] The data is currently not publicly available, though researchers may apply to Cambridge English for access to the Cambridge Learner Corpus, a collection of written essays.

[5] Note that full installation instructions for Redshift are provided at `http://russellmoo.re/cs/redshift`.

within a sentence. This method of representing grammatical relations is now widely-used in the NLP community for parser evaluation and owes much to the framework of Lexical-Functional Grammar [4].

In brief, each dependency is a triple (t, h, r): t is a token in the sentence, h is a token that is the 'head' or 'governor' of t, and r is a relation label showing how t modifies h (*e.g. nsubj* for nominal subject, *aux* for auxiliary). It is common to represent this as a directed labelled arc joining the two tokens: $h \rightarrow_{\mathrm{r}} t$.

Each token must have a single head, and the relations are constrained to form a projective tree headed by a single word, which itself is headed by a special ROOT token. To mark tokens that have been removed from the sentence by an EDIT transition, H&J added the *erased* relation to the SD set (see also [22, 23]). In these cases, the token is self-governing: $t \rightarrow_{\mathrm{erased}} t$.

The task, then, is to correctly identify any non-erased token's head: the unlabelled attachment score (UAS) is a measure of success in this respect[6].

Task 2: Disfluency detection We follow the evaluation defined by Charniak & Johnson [10], according to which, out of all the disfluency types, only the reparandum (RM) is the target for automatic detection. The reasoning behind this is that the filled pause (FP) and interregnum (IM) are said to be straightforwardly identified with a rule-based approach[7], whereas the RM is what needs to be *edited* out for a successful parse of the repaired string (RP). The task, then, is to successfully apply an EDIT transition to RM tokens, and results are presented as the disfluency F-measure (Disfl.F) – a computation over precision (p) and recall (r) as follows: $F = 2 \times \frac{p \times r}{p + r}$

4 Results

The performance of the Redshift parser in the two tasks described in section 3 is given in Table 1. **UAS** is a measure of parse accuracy, indicating the percentage of correctly-identified dependency-head relations. **Disfl.F** represents accuracy at disfluency detection, being the harmonic mean of precision **Disfl.P** and recall **Disfl.R** on this task.

Results are reported for various test-sets as follows: **SWB-SYN-DISF:TEST**[8] represents the averaged results from [16], with Redshift trained on the 600k token SWB-SYN-DISF dataset (90.5% UAS, 84.1% Disfl.F).

[6] Following [16] we do not report the labelled attachment score (LAS) though acknowledge that this would be an interesting direction for future work.

[7] To our knowledge these rules have not been codified, presumably because it is assumed a trivial task. We assume FP detection would rely heavily on UH POS tags, which are often accurately applied – though not always, an issue briefly discussed by Caines & Buttery [8]. We assume that a bigram rule would account for the IM, or 'parenthetical', which is usually exemplified as either 'you know' or 'I mean'; but we have similar concerns here, as it is not clear that distinguishing IMs from subordinating uses of these chunks (*e.g.* 'I mean what I say'; 'you know what I mean') is as straightforward as it is presumed to be.

[8] Following standard practice in the disfluency detection literature, the train/dev/test splits are those described in [10].

	Treebank	Sentences	Tokens	UAS	Disfl.P/R	Disfl.F
native	SWB-SYN-DISF:TEST	3900	45,405	90.5	n/a	84.1
	SWB-SYN-DISF:DEV	3833	45,381	90.9	92.3/76.5	83.7
non-native	BULATS:ALL	381	5667	91.1	82.6/33.6	47.8
	BULATS:B1	121	1895	88.9	85.3/31.4	45.9
	BULATS:B2	136	1879	91.2	79.2/33.2	46.8
	BULATS:C1	124	1893	93.0	83.8/37.3	51.6

Table 1. Redshift parse (UAS) and disfluency (Disfl.P/R, Disfl.F) accuracies on the Switchboard and BULATS test-sets.

To verify Redshift's performance figures versus other parsers, H&J averaged results across twenty randomly-seeded training runs. To compare datasets, we chose to keep the parsing model constant, using a single training run with the default perceptron seeding. We benchmarked the parser on **SWB-SYN-DISF:DEV** and observed a near-identical score to H&J (90.9% UAS, 83.7% Disfl.F).

From our BULATS non-native speaker corpus, we have results for the CEFR level subcorpora – **BULATS:B1**, **BULATS:B2**, **BULATS:C1** – as well as for the combined BULATS treebank, **BULATS:ALL**. For the combined data, UAS is similar to the scores obtained on the SWB corpora (91.1%) but Disfl.F is markedly reduced (47.8%). In section 5 we discuss this result, but the major factor seems to be a far lower recall rate – many disfluencies in the BULATS data are simply being missed. As for the outcome by learner proficiency, both UAS and Disf.F improve with increasing CEFR level. This indicates that the learners' speech approximates something like native speech as proficiency increases, thus both dependency relations and disfluencies are more accurately identified. We suggest that this trend makes parsing a useful diagnostic in ALA and CALL applications.

5 Discussion

Any CALL or ALA applications rely upon an accurate understanding of natural language, within which analysis of language relations in the *who did what to whom* sense are a crucial component. Spoken language is inherently disfluent and presents an acute challenge in attempts to identify these relations. As discussed here and in earlier work [16, 26, 27], a disfluent utterance may yet be appropriately parsed if the disfluent sections can be detected and removed. We have shown that the models of disfluency detection developed for native speaker data enjoy less success with learner data, at least of the kind presented here (business exam monologues). This means that our automated understanding of what the learner meant to say is impaired, with negative implications for CALL and ALA models.

We observe both a lower precision and a notably lower recall rate than is attained when parsing native speech. The implication is that in BULATS the errors are slightly harder to correct, but much harder to detect, than in SWB.

In comparison to the native speaker's FP-RM-IM-RR sequence given in Figure 1, our non-native speaker disfluencies are rarely so orderly. Filled pauses (FP) were omit-

ted from the datasets used here, and interestingly the BULATS corpus has no clear interregna (IM) structures. 'Rough copy' edits (exact repetition, or repetition with insertion, deletion or substitution of one or more words [18]) are common in BULATS (131 of 207 disfluent sections – 63.3% – have exact repetition of at least one token between reparandum and repair) and are accommodated in Redshift with specific contextual devices that search the buffer and stack for nearby POS or word matches [16].

The remainder of the BULATS disfluencies are characterised by errors: incorrect lexical choices that are intitiated but subsequently abandoned – so-called 'false starts'. Of the 207 disfluent sections, 55 (26.6%) are speech errors of this kind. Many false starts are single token mispronunciations – *e.g.* 'health' followed by the corrective 'help', 'far' then 'fast', 'ship' then 'sitting'. These 'soundalikes' represent a challenge for disfluency detection systems that, without audio, cannot recognise them as a kind of repetition.

Many of the uncorrected reparanda in BULATS are long and not obviously related to the repair - often resembling complete grammatical structures. Some examples are given in Figure 2:

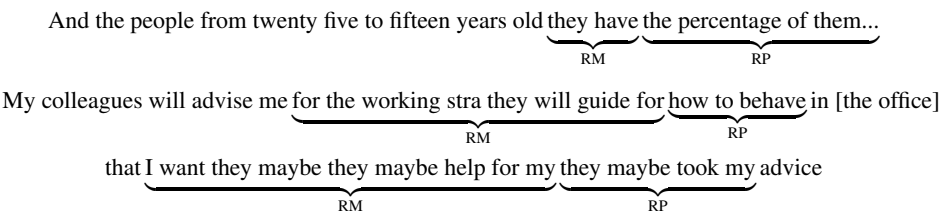

Fig. 2. Undetected or partially detected disfluency examples from the BULATS corpus.

Our future work involves further investigation of these errors and how they might automatically be recognised. We propose that these errors distinguish non-native speaker data (*e.g.* BULATS) from native speaker data (*e.g.* SWB) and that they are harder for language models to detect than rough copy, as reflected in the Disfl.F disparity.

Moreover, if disfluencies such as these can be detected more successfully in learner data, we envisage that a measure of the quantity of linguistic material edited out of the utterance will provide some measure of *distance* to canonical language, or the learner's 'target', whatever that is taken to be ('correct' standard English, as a first approximation). This, along with a measure of error correction, fits in with the ideas expressed in [8], in which increasing (normalised) parse tree probabilities from the RASP System were taken as a proxy for distance to the native speaker data on which the parser had been trained. The next step in this line of work is to develop a more appropriate model of disfluencies for non-native speaker data, that can accommodate the type of disfluency exemplified in Figure 2.

We acknowledge that at 5667 tokens in total, our BULATS dataset is dwarfed by the SWB corpus and its 45k dev/test-sets. However, preparation of a gold-standard treebank is a laborious task and has already taken many hours of work. However, new techniques – such as crowdsourcing as in [7] – are becoming available, and we intend to continue

expanding the BULATS treebank, in particular with speakers of other first languages and CEFR levels. We will also collect new data for speech topics other than business, for spontaneous dialogues as well as monologues, and for tasks other than oral examinations.

6 Acknowledgements

This paper reports on research supported by Cambridge English, University of Cambridge. We thank Ted Briscoe, Nick Saville, Fiona Barker, Ardeshir Geranpayeh, Nahal Khabbazbashi, and Matthew Honnibal for their advice and assistance, as well as the three anonymous reviewers for their helpful feedback.

References

1. Ballesteros, M., Nivre, J.: Going to the roots of dependency parsing. Computational Linguistics 39(1) (2013)
2. Biber, D.: Dimensions of register variation: a cross-linguistic comparison. Cambridge: Cambridge University Press (1995)
3. Brazil, D.: A grammar of speech. Oxford: Oxford University Press (1995)
4. Bresnan, J.: Lexical-Functional Syntax. Oxford: Blackwell (2001)
5. Briscoe, T., Carroll, J., Watson, R.: The second release of the RASP System. In: Proceedings of the COLING/ACL 2006 Interactive Presentations Session. Association for Computational Linguistics (2006)
6. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X). Association for Computational Linguistics (2006)
7. Caines, A., Bentz, C., Graham, C., Polzehl, T., Buttery, P.: Crowdsourcing a multi-lingual speech corpus: recording, transcription, and natural language processing. In: Proceedings of INTERSPEECH 2015. International Speech Communication Association (2015)
8. Caines, A., Buttery, P.: The effect of disfluencies and learner errors on the parsing of spoken learner language. In: Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages (2014)
9. Carter, R., McCarthy, M.: Spoken Grammar: where are we and where are we going? Applied Linguistics (in press)
10. Charniak, E., Johnson, M.: Edit detection and parsing for transcribed speech. In: Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL). Association for Computational Linguistics (2001)
11. Council of Europe: Common European Framework of Reference for Languages. Cambridge: Cambridge University Press (2001)
12. Foster, J., Çetinoğlu, Ö., Wagner, J., Roux, J.L., Nivre, J., Hogan, D., van Genabith, J.: From news to comment: resources and benchmarks for parsing the language of Web 2.0. In: Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP). Association for Computational Linguistics (2011)
13. Godfrey, J., Holliman, E.: Switchboard-1 Release 2 LDC97S62. DVD (1993)
14. Godfrey, J.J., Holliman, E.C., McDaniel, J.: SWITCHBOARD: telephone speech corpus for research and development. In: Proceedings of Acoustics, Speech, and Signal Processing (ICASSP-92). IEEE (1992)

15. Honnibal, M., Goldberg, Y., Johnson, M.: A non-monotonic arc-eager transition system for dependency parsing. In: Proceedings of the Seventh Conference on Computational Natural Language Learning. Association for Computational Linguistics (2013)
16. Honnibal, M., Johnson, M.: Joint incremental disfluency detection and dependency parsing. Transactions of the Association for Computational Linguistics 2, 131–142 (2014)
17. Hovy, D., Plank, B., Søgaard, A.: When POS data sets don't add up: combatting sample bias. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC). European Language Resources Association (2014)
18. Johnson, M., Charniak, E.: A TAG-based noisy channel model of speech repairs. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics (2004)
19. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics (2003)
20. Kübler, S., McDonald, R., Nivre, J.: Dependency parsing. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2009)
21. Lease, M., Charniak, E.: Parsing biomedical literature. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP). Association for Computational Linguistics (2005)
22. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC). European Language Resources Association (2006)
23. de Marneffe, M.C., Manning, C.D.: The Stanford typed dependencies representation. In: Proceedings of the COLING Workshop on Cross-framework and Cross-domain Parser Evaluation (2008)
24. Mikheev, A.: Text segmentation. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics. Oxford: Oxford University Press (2005)
25. Nivre, J.: Algorithms for deterministic incremental dependency parsing. Computational Linguistics 34(4), 513–553 (2008)
26. Qian, X., Liu, Y.: Disfluency detection using multi-step stacked learning. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Association for Computational Linguistics (2013)
27. Rasooli, M.S., Tetreault, J.: Non-monotonic parsing of *Fluent umm I Mean* disfluent sentences. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014). Association for Computational Linguistics (2014)
28. Rimell, L., Clark, S.: Porting a lexicalized-grammar parser to the biomedical domain. Journal of Biomedical Informatics 42, 852–865 (2009)
29. Shriberg, E.: Preliminaries to a theory of speech disfluencies. Ph.D. thesis, University of California, Berkeley (1994)
30. Taylor, A., Marcus, M., Santorini, B.: The Penn Treebank: An Overview (2003)
31. Zhang, Y., Clark, S.: Syntactic processing using the generalized perceptron and beam search. Computational Linguistics 37(1), 105–151 (2011)
32. Zhang, Y., Nivre, J.: Transition-based dependency parsing with rich non-local features. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT). Association for Computational Linguistics (2011)