The IIT Bombay Hindi⇔English Translation System at WMT 2014

Piyush Dungarwal, Rajen Chatterjee, Abhijit Mishra, Anoop Kunchukuttan, Ritesh Shah, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
{piyushdd,rajen,abhijitmishra,anoopk,ritesh,pb}@cse.iitb.ac.in

Abstract

In this paper, we describe our English-Hindi and Hindi-English statistical systems submitted to the WMT14 shared task. The core components of our translation systems are phrase based (Hindi-English) and factored (English-Hindi) SMT systems. We show that the use of number, case and Tree Adjoining Grammar information as factors helps to improve English-Hindi translation, primarily by generating morphological inflections correctly. We show improvements to the translation systems using pre-procesing and post-processing components. To overcome the structural divergence between English and Hindi, we preorder the source side sentence to conform to the target language word order. Since parallel corpus is limited, many words are not translated. We translate out-of-vocabulary words and transliterate named entities in a post-processing stage. We also investigate ranking of translations from multiple systems to select the best translation.

1 Introduction

India is a multilingual country with Hindi being the most widely spoken language. Hindi and English act as *link languages* across the country and languages of official communication for the Union Government. Thus, the importance of English⇔Hindi translation is obvious. Over the last decade, several rule based (Sinha, 1995) , interlingua based (Dave et. al., 2001) and statistical methods (Ramanathan et. al., 2008) have been explored for English-Hindi translation.

In the WMT 2014 shared task, we undertake the challenge of improving translation between the English and Hindi language pair using Statistical Machine Translation (SMT) techniques. The WMT 2014 shared task has provided a standardized test set to evaluate multiple approaches and avails the largest publicly downloadable English-Hindi parallel corpus. Using these resources, we have developed a phrase-based and a factored based system for Hindi-English and English-Hindi translation respectively, with pre-processing and post-processing components to handle structural divergence and morphlogical richness of Hindi. Section 2 describes the issues in Hindi↔English translation.

The rest of the paper is organized as follows. Section 3 describes corpus preparation and experimental setup. Section 4 and Section 5 describe our English-Hindi and Hindi-English translation systems respectively. Section 6 describes the post-processing operations on the output from the core translation system for handling OOV and named entities, and for reranking outputs from multiple systems. Section 7 mentions the details regarding our systems submitted to WMT shared task. Section 8 concludes the paper.

2 Problems in Hindi⇔English Translation

Languages can be differentiated in terms of structural divergences and morphological manifestations. English is structurally classified as a Subject-Verb-Object (SVO) language with a poor morphology whereas Hindi is a morphologically rich, Subject-Object-Verb (SOV) language. Largely, these divergences are responsible for the difficulties in translation using a phrase based/factored model, which we summarize in this section.

2.1 English-to-Hindi

The fundamental structural differences described earlier result in large distance verb and modifier movements across English-Hindi. Local reordering models prove to be inadequate to over-

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 90–96,
Baltimore, Maryland USA, June 26–27, 2014.

come the problem; hence, we transformed the source side sentence using pre-ordering rules to conform to the target word order. Availability of robust parsers for English makes this approach for English-Hindi translation effective.

As far as morphology is concerned, Hindi is more richer in terms of case-markers, inflection-rich surface forms including verb forms etc. Hindi exhibits gender agreement and syncretism in inflections, which are not observed in English. We attempt to enrich the source side English corpus with linguistic factors in order to overcome the morphological disparity.

2.2 Hindi-to-English

The lack of accurate linguistic parsers makes it difficult to overcome the structural divergence using preordering rules. In order to preorder Hindi sentences, we build rules using shallow parsing information. The source side reordering helps to reduce the decoder's search complexity and learn better phrase tables. Some of the other challenges in generation of English output are: (1) generation of articles, which Hindi lacks, (2) heavy overloading of English prepositions, making it difficult to predict them.

3 Experimental Setup

We process the corpus through appropriate filters for normalization and then create a train-test split.

3.1 English Corpus Normalization

To begin with, the English data was tokenized using the Stanford tokenizer (Klein and Manning, 2003) and then true-cased using *truecase.perl* provided in MOSES toolkit.

3.2 Hindi Corpus Normalization

For Hindi data, we first normalize the corpus using NLP Indic Library (Kunchukuttan et. al., 2014)[1]. Normalization is followed by tokenization, wherein we make use of the *trivtokenizer.pl*[2] provided with WMT14 shared task. In Table 1, we highlight some of the post normalization statistics for en-hi parallel corpora.

[1] https://bitbucket.org/anoopk/indic_nlp_library

[2] http://ufallab.ms.mff.cuni.cz/~bojar/hindencorp/

	English	**Hindi**
Token	2,898,810	3,092,555
Types	95,551	118,285
Total Characters	18,513,761	17,961,357
Total sentences	289,832	289,832
Sentences (word count $\leq$ 10)	188,993	182,777
Sentences (word count $>$ 10)	100,839	107,055

Table 1: en-hi corpora statistics, post normalisation.

3.3 Data Split

Before splitting the data, we first randomize the parallel corpus. We filter out English sentences longer than 50 words along with their parallel Hindi translations. After filtering, we select 5000 sentences which are 10 to 20 words long as the test data, while remaining 284,832 sentences are used for training.

4 English-to-Hindi (en-hi) translation

We use the MOSES toolkit (Koehn et. al., 2007a) for carrying out various experiments. Starting with Phrase Based Statistical Machine Translation (PBSMT)(Koehn et. al., 2003) as baseline system we go ahead with pre-order PBSMT described in Section 4.1. After pre-ordering, we train a Factor Based SMT(Koehn, 2007b) model, where we add factors on the pre-ordered source corpus. In Factor Based SMT we have two variations- (a) using *Supertag* as factor described in Section 4.2 and (b) using *number, case* as factors described in Section 4.3.

4.1 Pre-ordering source corpus

Research has shown that pre-ordering source language to conform to target language word order significantly improves translation quality (Collins et. al, 2005). There are many variations of pre-ordering systems primarily emerging from either rule based or statistical methods. We use rule based pre-ordering approach developed by (Patel et. al., 2013), which uses the Stanford parser (Klein and Manning, 2003) for parsing English sentences. This approach is an extension to an earlier approach developed by (Ramanathan et. al., 2008). The existing source reordering system requires the input text to contain only surface form, however, we extended it to support surface form

along with its factors like POS, lemma etc.. An example of improvement in translation after pre-ordering is shown below:
Example: trying to replace bad ideas with good ideas .
Phr: replace बुरे विचारों को अच्छे विचारों के साथ
(*replace bure vichaaron ko acche vichaaron ke saath*)
Gloss: replace bad ideas good ideas with
Pre-order PBSMT: अच्छे विचारों से बुरे विचारों को बदलने की कोशिश कर रहे हैं
(*acche vichaaron se bure vichaaron ko badalane ki koshish kara rahe hain*)
Gloss: good ideas with bad ideas to replace trying

4.2 Supertag as Factor

The notion of *Supertag* was first proposed by Joshi and Srinivas (1994). Supertags are elementary trees of Lexicalized Tree Adjoining Grammar (LTAG) (Joshi and Schabes, 1991). They provide syntactic as well as dependency information at the word level by imposing complex constraints in a local context. These elementary trees are combined in some manner to form a parse tree, due to which, supertagging is also known as "An approach to almost parsing"(Bangalore and Joshi, 1999). A supertag can also be viewed as fragments of parse trees associated with each lexical item. Figure 1 shows an example of supertagged sentence "The purchase price includes taxes"described in (Hassan et. al., 2007). It clearly shows the sub-categorization information available in the verb *include*, which takes subject NP to its left and an object NP to its right.

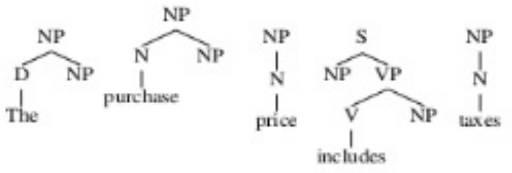

Figure 1: LTAG supertag sequence obtained using MICA Parser.

Use of supertags as factors has already been studied by Hassan (2007) in context of Arabic-English SMT. They use supertag language model along with supertagged English corpus. Ours is the first study in using supertag as factor for English-to-Hindi translation on a pre-ordered source corpus.

We use MICA Parser (Bangalore et. al., 2009) for obtaining supertags. After supertagging we run pre-ordering system preserving the supertags in it. For translation, we create mapping from *source-word|supertag* to *target-word*. An example of improvement in translation by using supertag as factor is shown below:
Example: trying to understand what your child is saying to you
Phr: आपका बच्चा आपसे क्या कह रहा है यह
(*aapkaa bacchaa aapse kya kaha rahaa hai yaha*)
Gloss: your child you what saying is this
Supertag Fact: आपका बच्चा आपसे क्या कह रहा है , उसे समझने की कोशिश करना
(*aapkaa bacchaa aapse kya kaha rahaa hai, use samajhane kii koshish karnaa*)
Gloss: your child to you what saying is , that understand try

4.3 Number, Case as Factor

In this section, we discuss how to generate correct noun inflections while translating from English to Hindi. There has been previous work done in order to solve the problem of *data sparsity* due to complex *verb morphology* for English to Hindi translation (Gandhe, 2011). Noun inflections in Hindi are affected by the number and case of the noun only. *Number* can be singular or plural, whereas, *case* can be direct or oblique. We use the factored SMT model to incorporate this linguistic information during training of the translation models. We attach *root-word*, *number* and *case* as factors to English nouns. On the other hand, to Hindi nouns we attach *root-word* and *suffix* as factors. We define the translation and generation step as follows:

- Translation step (T0): Translates English $root|number|case$ to Hindi $root|suffix$
- Generation step (G0): Generates Hindi surface word from Hindi $root|suffix$

An example of improvement in translation by using number and case as factors is shown below:
Example: Two sets of statistics
Phr: दो के आँकड़े
(*do ke aankade*)
Gloss: two of statistics
Num-Case Fact: आँकड़ों के दो सेट
(*aankadon ke do set*)
Gloss: statistics of two sets

4.3.1 Generating number and case factors

With the help of syntactic and morphological tools, we extract the number and case of the English nouns as follows:

- **Number factor:** We use *Stanford POS tagger*[3] to identify the English noun entities (Toutanova, 2003). The POS tagger itself differentiates between singular and plural nouns by using different tags.
- **Case factor:** It is difficult to find the direct/oblique case of the nouns as English nouns do not contain this information. Hence, to get the case information, we need to find out features of an English sentence that correspond to direct/oblique case of the parallel nouns in Hindi sentence. We use object of preposition, subject, direct object, tense as our features. These features are extracted using semantic relations provided by Stanford's typed dependencies (Marneffe, 2008).

4.4 Results

Listed below are different statistical systems trained using *Moses*:

- Phrase Based model (*Phr*)
- Phrase Based model with pre-ordered source corpus (*PhrReord*)
- Factor Based Model with factors on pre-ordered source corpus
  - Supertag as factor (*PhrReord+STag*)
  - Number, Case as factor (*PhrReord+NC*)

We evaluated translation systems with BLEU and TER as shown in Table 2. Evaluation on the development set shows that factor based models achieve competitive scores as compared to the baseline system, whereas, evaluation on the WMT14 test set shows significant improvement in the performance of factor based models.

	Development		WMT14	
Model	**BLEU**	**TER**	**BLEU**	**TER**
Phr	27.62	0.63	8.0	0.84
PhrReord	28.64	0.62	8.6	0.86
PhrReord+STag	27.05	0.64	9.8	0.83
PhrReord+NC	27.50	0.64	10.1	0.83

Table 2: English-to-Hindi automatic evaluation on development set and on WMT14 test set.

5 Hindi-to-English (hi-en) translation

As English follows SVO word order and Hindi follows SOV word order, simple distortion penalty in phrase-based models can not handle the reordering well. For the shared task, we follow the approach that pre-orders the source sentence to conform to target word order.

A substantial volume of work has been done in the field of source-side reordering for machine translation. Most of the experiments are based on applying reordering rules at the nodes of the parse tree of the source sentence. These reordering rules can be automatically learnt (Genzel, 2010). But, many source languages do not have a good robust parser. Hence, instead we can use shallow parsing techniques to get chunks of words and then reorder them. Reordering rules can be learned automatically from chunked data (Zhang, 2007).

Hindi does not have a functional constituency or dependency parser available, as of now. But, a shallow parser[4] is available for Hindi. Hence, we follow a chunk-based pre-ordering approach, wherein, we develop a set of rules to reorder the chunks in a source sentence. The following are the chunks tags generated by this shallow parser: Noun chunks (NP), Verb chunks (VGF, VGNF, VGNN), Adjectival chunks (JJP), Adverb chunks (RBP), Negatives (NEGP), Conjuncts (CCP), Chunk fragments (FRAGP), and miscellaneous entities (BLK) (Bharati, 2006).

5.1 Development of rules

After chunking an input sentence, we apply handcrafted reordering rules on these chunks. Following sections describe these rules. Note that we apply rules in the same order they are listed below.

5.1.1 Merging of chunks

After chunking, we merge the adjacent chunks, if they follow same order in target language.

1. Merge {JJP VGF} chunks (Consider this chunk as a single VGF chunk)
   e.g., वर्णित है (*varnit hai*), स्थित है (*sthit hai*)

[3] http://nlp.stanford.edu/software/tagger.shtml

[4] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

2. Merge adjacent verb chunks (Consider this chunk as a single verb chunk)
   e.g., गिरता है (*girataa hai*), लुभाता है (*lubhaataa hai*)

3. Merge NP and JJP chunks separated by commas and CCP (Consider this chunk as a single NP chunk)
   e.g., बड़ा और अहम (*badaa aur aham*)

5.1.2 Preposition chunk reordering

Next we find sequence of contiguous chunks separated by prepositions (Can end in verb chunks). We apply following reordering rules on these contiguous chunks:

1. Reorder multi-word preposition locally by reversing the order of words in that chunk
   e.g., के अलावा (*ke alaawaa*) → अलावा के, के सामने (*ke saamane*) → सामने के

2. Reorder contiguous preposition chunk by reversing the order of chunks (Consider this chunk as a single noun chunk)
   e.g., हिंदू धर्म में तीर्थ का बड़ा महत्व (*hinduu dharma me tirtha ka badaa mahatva*) → बड़ा महत्व का तीर्थ में हिंदू धर्म

5.1.3 Verb chunk reordering

We find contiguous verb chunks and apply following reordering rules:

1. Reorder chunks locally by reversing the order of the chunks
   e.g., वर्णित है (*varnit hai*) → है वर्णित

2. Verb chunk placement: We place the new verb chunk after first NP chunk. Same rule applies for all verb chunks in a sentence, i.e., we place each verb chunk after first NP chunk of the clause to which the verb belongs.

   Note that, even though placing verb chunk after first NP chunk may be wrong reordering. But we also use distortion window of 6 to 20 while using phrase-based model. Hence, further reordering of verb chunks can be somewhat handled by phrase-based model itself.

Thus, using chunker and reordering rules, we get a source-reordered Hindi sentence.

5.2 Results

We trained two different translation models:

- Phrase-based model without source reordering (*Phr*)
- Phrase-based model with chunk-based source reordering (*PhrReord*)

	Development		WMT14	
Model	**BLEU**	**TER**	**BLEU**	**TER**
Phr	27.53	0.59	13.5	0.87
PhrReord	25.06	0.62	13.7	0.90

Table 3: Hindi-to-English automatic evaluation on development set and on WMT14 test set.

Table 3 shows evaluation scores for development set and WMT14 test set. Even though we do not see significant improvement in automatic evaluation of *PhrReord*, but this model contributes in improving translation quality after ranking, as discussed in Section 5. In subjective evaluation we found many translation to be better in *PhrReord* model as shown in the following examples:

Example 1: सन 2004 से वे कई बार चोटग्रस्त रहे हैं |
(*sana 2004 se ve kaii baar chotagrasta rahe hain.*)
Phr: *since 2004 he is injured sometimes .*
PhrReord: *he was injured many times since 2004 .*

Example 2: ओबामा का राष्ट्रपति पद के चुनाव प्रचार हेतु बनाया आधिकारिक जालस्थल
(*obama ka rashtrapti pad ke chunaav prachaar hetu banaayaa aadhikarik jaalsthal*)
Phr: of Obama for election campaign
PhrReord: official website of Obama created for President campaign

6 Post processing

All experimental results reported in this paper are after post processing the translation output. In post processing, we remove some Out-of-Vocabulary (OOV) words as described in subsection 6.1, after which we transliterate the remaining OOV words.

6.1 Removing OOV

We noticed, there are many words in the training corpus which were not present in the phrase table, but, were present in the lexical tranlsation table. So we used the lexical table as a dictionary to lookup bilingual translations. Table 4 gives the statistics of number of OOV reduced.

Model	Before	After
Phrased Based	2313	1354
Phrase Based (pre-order)	2256	1334
Supertag as factor	4361	1611
Num-Case as factor	2628	1341

Table 4: Statistics showing number of OOV before and after post processing the English-to-Hindi translation output of Development set.

6.2 Transliteration of Untranslated Words

OOV words which were not present in the lexical translation table were then transliterated using a naive transliteration system. The transliteration step was applied on Hindi-to-English translation outputs only. After transliteration we noticed fractional improvements in BLEU score varying from 0.1 to 0.5.

6.3 Ranking of Ensemble MT Output

We propose a ranking framework to select the best translation output from an ensemble of multiple MT systems. In order to exploit the strength of each system, we augment the translation pipeline with a ranking module as a post processing step. For English-to-Hindi ranking we combine the output of both factor based models, whereas, for Hindi-to-English ranking we combine *phrase based* and *phrase based with pre-ordering* outputs.

For most of the systems, the output translations are adequate but not fluent enough. So, based on their fluency scores, we decided to rank the candidate translations. Fluency is well quantified by *LM log probability score* and *Perplexity*. For a given translation , we compute these scores by querying the 5-gram language model built using SRILM. Table 5 shows more than 4% relative improvement in BLEU score for en-hi as well as hi-en translation system after applying ranking module.

Model	BLEU	METEOR	TER
Phr(en-hi)	27.62	0.41	0.63
After Ranking (en-hi)	28.82	0.42	0.63
Phr(hi-en)	27.53	0.27	0.59
After Ranking (hi-en)	28.69	0.27	0.59

Table 5: Comparision of ranking score with baseline

7 Primary Systems in WMT14

For English-to-Hindi, we submitted the ranked output of factored models trained on pre-ordered source corpus. For Hindi-to-English, we submitted the ranked output of phrase based and pre-ordered phrase based models. Table 6 shows evaluation scores of these systems on WMT14 test set.

Lang. pair	BLEU	TER
en-hi	10.4	0.83
hi-en	14.5	0.89

Table 6: WMT14 evaluation for *en-hi* and *hi-en*.

8 Conclusion

We conclude that the difficulties in English-Hindi MT can be tackled by the use of factor based SMT and various pre-processing and post processing techniques. Following are our primary contributions towards English-Hindi machine translation:

- Use of supertag factors for better translation of structurally complex sentences
- Use of number-case factors for accurately generating noun inflections in Hindi
- Use of shallow parsing for pre-ordering Hindi source corpus

We also observed that simple ranking strategy benefits in getting the best translation from an ensemble of translation systems.

References

Avramidis, Eleftherios, and Philipp Koehn. 2008. *Enriching Morphologically Poor Languages for Statistical Machine Translation*. ACL.

Banerjee, Satanjeev, and Alon Lavie. 2005. *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments*. Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization.

Srinivas Bangalore and Aravind K. Joshi. 1999. *Supertagging: An approach to almost parsing*. Computational linguistics.

Srinivas Bangalore, Pierre Boulllier, Alexis Nasr, Owen Rambow, and BenoÃőt Sagot. 2009. *MICA: a probabilistic dependency parser based on tree insertion grammars application note*. Proceedings of

Human Language Technologies The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics.

A. Bharati, R. Sangal, D. M. Sharma and L. Bai. 2006. *AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages*. Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad.

Dave, Shachi and Parikh, Jignashu and Bhattacharyya, Pushpak. 2001. *Interlingua-based English–Hindi Machine Translation and Language Divergence* Journal Machine Translation

Gandhe, Ankur, Rashmi Gangadharaiah, Karthik Visweswariah, and Ananthakrishnan Ramanathan. 2011. *Handling verb phrase morphology in highly inflected Indian languages for Machine Translation*. IJCNLP.

Genzel, Dmitriy. 2010. *Automatically learning source-side reordering rules for large scale machine translation* Proceedings of the 23rd international conference on computational linguistics. Association for Computational Linguistics

Hany Hassan, Khalil Sima'an, and Andy Way 2007. *Supertagged phrase-based statistical machine translation*. Proceedings of the Association for Computational Linguistics Association for Computational Linguistics.

Aravind K. Joshi and Yves Schabes 1991. *Tree-adjoining grammars and lexicalized grammars*. Technical Report No. MS-CIS-91-22

Dan Klein and Christopher D. Manning. 2003. *Accurate Unlexicalized Parsing*. Proceedings of the 41st Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. *Statistical phrase-based translation.* Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, OndÅŹej Bojar, Alexandra Constantin and Evan Herbst. 2007. *Moses: open source toolkit for statistical machine translation*. Proceedings of the Second Workshop on Hybrid Approaches to Translation. Association for Computational Linguistics.

Philipp Koehn and Hieu Hoang 2007. *Factored Translation Models* Conference on Empirical Methods in Natural Language Processing.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee,Ritesh Shah, and Pushpak Bhattacharyya. 2014. *Sata-Anuvadak: Tackling Multiway Translation of Indian Languages*. Proceedings of the Ninth International Conference on Language Resources and Evaluation Conference

De Marneffe, Marie-Catherine, and Christopher D. Manning. 2008. *Stanford typed dependencies manual*. URL http://nlp. stanford. edu/software/dependencies manual. pdf (2008).

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics.

Raj Nath Patel, Rohit Gupta, Prakash B. Pimpale and Sasikumar M. 2013. *Reordering rules for English-Hindi SMT*. Proceedings of the Second Workshop on Hybrid Approaches to Translation. Association for Computational Linguistics.

Ananthakrishnan Ramanathan, Pushpak Bhattacharyya, Jayprasad Hegde, Ritesh M. Shah, and M. Sasikumar. 2008. *Simple syntactic and morphological processing can help English-Hindi statistical machine translation*. In International Joint Conference on NLP.

Sinha, RMK and Sivaraman, K and Agrawal, A and Jain, R and Srivastava, R and Jain, A. 1995. *ANGLABHARTI: a multilingual machine aided translation project on translation from English to Indian languages* IEEE International Conference on Systems, Man and Cybernetics

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer 2003. *Feature-rich part-of-speech tagging with a cyclic dependency network*. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics.

Zhang, Yuqi, Richard Zens, and Hermann Ney. 2007. *Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation* Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation. Association for Computational Linguistics

Collins, Michael, Philipp Koehn, and Ivona Kučerova 2005 *Clause restructuring for statistical machine translation.* Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics