Manual Annotations for Evaluating the Readability of Automatically Produced Summaries

This file is a supplementary material for our paper “*Group: The Better Language Unit for Extractive Text Summarization*” which has not published yet. We randomly select 18 long papers and 17 short papers from the *ACL2014* dataset, and automatically produce the *sentence*-composed, *group-*composed and *paragraph-*composedsummaries for these papers. Two volunteers A and B are invited to evaluate the readability of the produced summaries. The readability annotations for each produced summary include the dangling anaphors and the readability ratings. The dangling anaphors are the anaphors that cannot be correctly resolved in the produced summaries. The readability ratings range from 0 to 5. The 0-rating means the sentences in produced summaries are not coherent at all, and the whole text of the produced summary does not make any sense. While, the 5-rating means any two consecutive sentences in the produced summary are coherent, and the whole text of the produced summary has a complete meaning.

Table E-1 to Table E-35 list all the produced summaries in the *Sample* dataset and the readability annotations. The underlined words in bold are the dangling anaphors annotated by volunteers. The texts in the *Abstract*, that serve as standard summaries for limiting the length of the produced summaries and for calculating the ROUGE scores of the produced summaries, are also listed in tables to facilitate readers to get a general idea of each randomly selected papers. However, we did not show the texts in the *Abstract* to the two volunteers when they were annotating the dangling anaphors and the readability ratings. This is because the information in the *Abstract* will act as the volunteers’ background knowledge after they read the *Abstract*, and this background knowledge will lead the volunteers to miss some dangling anaphors. In addition, the readability ratings of the produced summaries are just about whether the sentences in the produced summaries are coherent and whether the meanings of the produced summaries are complete, and do not require the text in the produced summaries to be similar with the text in the *Abstract*. The similarity between the produced summaries and the standard summaries are evaluated by the ROUGE-based metrics. If the volunteers read the *Abstract,* they will unconsciously compare the produced summaries with the *Abstract*, and this will affect their ratings for the readability of the produced summaries.

Table E-1. Standard summary, produced summaries and manual readability annotations of P14-1007.

|  |  |  |
| --- | --- | --- |
| **P14-1007** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | In this work, we revisit Shared Task 1 from the 2012 \*SEM Conference: the automated analysis of negation. Unlike the vast majority of participating systems in 2012, our approach works over explicit and formal representations of propositional semantics, i.e. derives the notion of negation scope assumed in this task from the structure of logical-form meaning representations. We relate the task-specific interpretation of (negation) scope to the concept of (quantifier and operator) scope in mainstream underspecified semantics. With reference to an explicit encoding of semantic predicate-argument structure, we can operationalize the annotation decisions made for the 2012 \*SEM task, and demonstrate how a comparatively simple system for negation scope resolution can be built from an off-the-shelf deep parsing system. In a system combination setting, our approach improves over the best-published results on this task to date. |  |
| **Sentence-composed summary** | **s7**: **Example (1)**, where hi marks **the cue** and {} the in-scope elements, illustrates **the annotations**, including how negation inside a noun phrase can scope over discontinuous parts of the sentence.  **s87**: Thus, the **MRS** crawling operations ‘paint’ a subset of the MRS graph as in-scope for a given negation cue.  **s106**: Our MRS crawling algorithm was defined by looking at the annotated data rather than the annotation guidelines for **the Shared Task** [7].  **s119**: Since in **this case**, the negation cue will not be a quantifier in the MRS, there will be no functor crawling to **the verb**’s **EP**.  **s166**: We evaluated the performance of our system using the Shared Task development and evaluation data (respectively CDD and CDE in **Table 1**).  **s216**: Furthermore, even a system using syntactic structure to model **scope** would be faced with a more complicated task than our crawling rules. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors:  10 |
| **Group-composed summary** | **g57**: For negated subjects and objects, **the guidelines** state that the negation scopes over “all the clause” and “the clause headed by the verb” (Morante et al., 2011, 19), respectively. The examples given in the annotation guidelines suggest that these are in fact meant to refer to the same thing. The negation cue for a negated nominal argument will appear as a quantifier **EP** in the **MRS**, triggering line 3 of our algorithm. This functor crawling step will get to the verb’s EP, and from there, the process is the same as **the last two cases**.  **g70**: In contrast to subjects and objects, negation of a clausal argument is not treated as negation of the verb (ibid., p. 18). Since in this case, the negation cue will not be a **quantifier** in the MRS, there will be no functor crawling to the verb’s EP. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors:  5 |
| **Paragraph-composed summary** | **p27**: For negated subjects and objects, **the guidelines** state that the negation scopes over “all the clause” and “the clause headed by the verb” (Morante et al., 2011, 19), respectively. The examples given in the annotation guidelines suggest that these are in fact meant to refer to the same thing. The negation cue for a negated nominal argument will appear as a quantifier **EP** in the **MRS**, triggering line 3 of our algorithm. This functor crawling step will get to the verb’s EP, and from there, the process is the same as **the last two cases**.  **p36**: The close match between our crawling algorithm and the annotation guidelines supported by the mapping to MRS provides for very high precision and recall when **the analysis engine** produces the desired MRS. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors:  5 |

Table E-2. Standard summary, produced summaries and manual readability annotations of P14-1023.

|  |  |  |
| --- | --- | --- |
| **P14-1023** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Context-predicting models (more commonly known as embeddings or neural language models) are the new kids on the distributional semantics block. Despite the buzz surrounding these models, the literature is still lacking a systematic comparison of the predictive models with classic, count-vector-based distributional semantic approaches. In this paper, we perform such an extensive evaluation, on a wide range of lexical semantics tasks and across many parameter settings. The results, to our own surprise, show that the buzz is fully justified, as the context-predicting models obtain a thorough and resounding victory against their count-based counterparts. |  |
| **Sentence-composed summary** | **s134**: The first block of **the table** reports the maximum per-task performance (across all considered parameter settings) for **count and predict vectors**.  **s139**: The success of the predict models cannot be blamed on poor performance of the count models.  **s140**: Besides the fact that this would not explain the near-state-of-the-art performance of the predict vectors, the count model results are actually quite good in absolute terms.  **s142**: Interestingly, count vectors achieve performance comparable to that of predict vectors only on **the selectional preference tasks**. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 3 |
| **Group-composed summary** | **g42**: While **all the previous data sets** are relatively standard in **the DSM** field to test traditional count models, our last benchmark was introduced in Mikolov et al. 2013a ) specifically to test predict models.  **g**50: The success of the predict models cannot be blamed on poor performance of the count models. Besides the fact that this would not explain the near-state-of-the-art performance of the predict vectors, the count model results are actually quite good in absolute terms. Indeed, in several cases they are close, or even better than those attained by dm, a linguistically-sophisticated count-based approach that was shown to reach top performance across a variety of tasks by Baroni and Lenci ( 2010 ). | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 2 |
| **Paragraph-composed summary** | **p25**: The success of the predict models cannot be blamed on poor performance of the count models. Besides the fact that this would not explain the near-state-of-the-art performance of the predict vectors, the count model results are actually quite good in absolute terms. Indeed, in several cases they are close, or even better than those attained by dm, a linguistically-sophisticated count-based approach that was shown to reach top performance across a variety of tasks by Baroni and Lenci ( 2010 ) .  **p**29: Finally, we go back to **Table 2** to point out the poor performance of the out-of-the-box cw model. | **A**: 4 |
| **B**: 3 |
| Number of dangling anaphors: 1 |

Table E-3. Standard summary, produced summaries and manual readability annotations of P14-1039.

|  |  |  |
| --- | --- | --- |
| **P14-1039** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | We investigate whether parsers can be used for self-monitoring in surface realization in order to avoid egregious errors involving “vicious” ambiguities, namely those where the intended interpretation fails to be considerably more likely than alternative ones. Using parse accuracy in a simple reranking strategy for self-monitoring, we find that with a state-of-the-art averaged perceptron realization ranking model, BLEU scores cannot be improved with any of the well-known Treebank parsers we tested, since these parsers too often make errors that human readers would be unlikely to make. However, by using an SVM ranker to combine the realizer’s model score together with features from multiple parsers, including ones designed to make the ranker more robust to parsing mistakes, we show that significant increases in BLEU scores can be achieved. Moreover, via a targeted manual analysis, we demonstrate that the SVM reranker frequently manages to avoid vicious ambiguities, while its ranking errors tend to affect fluency much more often than adequacy |  |
| **Sentence-composed summary** | **s17**: Therefore, to develop a more nuanced self-monitoring reranker that is more robust to **such parsing mistakes**, we trained an SVM using dependency precision and recall features for **all three parses**, their n-best parsing results, and per-label precision and recall for each type of dependency, together with **the realizer**’s normalized perceptron model score as a feature.  **s77**: Similarly, we conjectured that large differences in the realizer’s perceptron model score may more reliably reflect human fluency preferences than small ones, and thus we combined this score with features for parser accuracy in an SVM ranker.  **s123**: In this paper, we have shown that while using parse accuracy in a simple reranking strategy for self-monitoring fails to improve BLEU scores over a state-of-the-art averaged perceptron realization ranking model, it is possible to significantly increase BLEU scores using an SVM ranker that combines the realizer’s model score together with features from multiple parsers, including ones designed to make the ranker more robust to parsing mistakes that human readers would be unlikely to make. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 3 |
| **Group-composed summary** | **g5**: Our simple reranking strategy for self-monitoring is to rerank **the realizer**’s n-best list by parse accuracy, preserving the original order in case of ties. In this way, if there is a realization in the n-best list that can be parsed more accurately than the top-ranked realization—even if the intended interpretation cannot be recovered with 100% accuracy—it will become the preferred output of the combined realization-with-self-monitoring system.  **g**39: In this paper, we have shown that while using parse accuracy in a simple reranking strategy for self-monitoring fails to improve BLEU scores over a state-of-the-art averaged perceptron realization ranking model, it is possible to significantly increase BLEU scores using an SVM ranker that combines the realizer’s model score together with features from multiple parsers, including ones designed to make the ranker more robust to parsing mistakes that human readers would be unlikely to make. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 1 |
| **Paragraph-composed summary** | **p5**: Our simple reranking strategy for self-monitoring is to rerank **the realizer**’s n-best list by parse accuracy, preserving the original order in case of ties. In this way, if there is a realization in the n-best list that can be parsed more accurately than the top-ranked realization—even if the intended interpretation cannot be recovered with 100% accuracy—it will become the preferred output of the combined realization-with-self-monitoring system.  **p**31: In this paper, we have shown that while using parse accuracy in a simple reranking strategy for self-monitoring fails to improve BLEU scores over a state-of-the-art averaged perceptron realization ranking model, it is possible to significantly increase BLEU scores using an SVM ranker that combines the realizer’s model score together with features from multiple parsers, including ones designed to make the ranker more robust to parsing mistakes that human readers would be unlikely to make. | **A**: 4 |
| **B**: 3 |
| Number of dangling anaphors: 1 |

Table E-4. Standard summary, produced summaries and manual readability annotations of P14-1064.

|  |  |  |
| --- | --- | --- |
| **P14-1064** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Statistical phrase-based translation learns translation rules from bilingual corpora, and has traditionally only used monolingual evidence to construct features that rescore existing translation candidates. In this work, we present a semi-supervised graph-based approach for generating new translation rules that leverages bilingual and monolingual data. The proposed technique first constructs phrase graphs using both source and target language monolingual corpora. Next, graph propagation identifies translations of phrases that were not observed in the bilingual corpus, assuming that similar phrases have similar translations. We report results on a large Arabic-English system and a medium-sized Urdu-English system. Our proposed approach significantly improves the performance of competitive phrasebased systems, leading to consistent improvements between 1 and 4 BLEU points on standard evaluation sets. |  |
| **Sentence-composed summary** | **s13**: We then limit **the set of translation options** for each unlabeled source phrase (§2.3), and using a structured graph propagation algorithm, where translation information is propagated from labeled to unlabeled phrases proportional to both source and target phrase similarities, we estimate probability distributions over translations for the unlabeled source phrases (§2.4).  **s67**: The morphologically-generated candidates for a given source unlabeled phrase are initially defined as the target word sequences in the monolingual data that have the same **stem sequence** as one of the baseline’s target translations for a source phrase which has the same stem sequence as the unlabeled source phrase.  **s74**: In our problem, the “label” for each node is actually a probability distribution over a set of translation candidates (target phrases). | **A**: 0 |
| **B**: 0 |
| Number of dangling anaphors: 2 |
| **Group-composed summary** | **g5**: We then limit **the set of translation options** for each unlabeled source phrase (§2.3), and using a structured graph propagation algorithm, where translation information is propagated from labeled to unlabeled phrases proportional to both source and target phrase similarities, we estimate probability distributions over translations for the unlabeled source phrases (§2.4).  **g**30: A graph propagation algorithm transfers label information from labeled nodes to unlabeled nodes by following the graph’s structure. In some applications, a label may consist of class membership information, e.g., each node can belong to one of a certain number of classes. In our problem, the “label” for each node is actually a probability distribution over a set of translation candidates (target phrases). | **A**: 4 |
| **B**: 5 |
| Number of dangling anaphors: 1 |
| **Paragraph-composed summary** | **p4**: Monolingual data is used to construct separate similarity graphs over phrases (word sequences), as illustrated in **Fig. 1**.  **p**19: A graph propagation algorithm transfers label information from labeled nodes to unlabeled nodes by following the graph’s structure. In some applications, a label may consist of class membership information, e.g., each node can belong to one of a certain number of classes. In our problem, the “label” for each node is actually a probability distribution over a set of translation candidates (target phrases). For a given node f, let e refer to a candidate in the label set for node f; then in graph propagation, the probability of candidate e given source phrase f in iteration t + 1 is: | **A**: 3 |
| **B**: 2 |
| Number of dangling anaphors: 1 |

Table E-5. Standard summary, produced summaries and manual readability annotations of P14-1067.

|  |  |  |
| --- | --- | --- |
| **P14-1067** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | The automatic estimation of machine translation (MT) output quality is a hard task in which the selection of the appropriate algorithm and the most predictive features over reasonably sized training sets plays a crucial role. When moving from controlled lab evaluations to real-life scenarios the task becomes even harder. For current MT quality estimation (QE) systems, additional complexity comes from the difficulty to model user and domain changes. Indeed, the instability of the systems with respect to data coming from different distributions calls for adaptive solutions that react to new operating conditions. To tackle this issue we propose an online framework for adaptive QE that targets reactivity and robustness to user and domain changes. Contrastive experiments in different testing conditions involving user and domain changes demonstrate the effectiveness of our approach. |  |
| **Sentence-composed summary** | **s30**: Our results show that the sensitivity of online **QE** models to different distributions of training and test instances makes them more suitable than batch methods for integration in a **CAT** framework.  **s87**: The batch model is built by learning only from the training data and is evaluated on the test set without exploiting information from the test instances.  **s166**: **To this aim**, our QE models are created using a training set coming from one domain (L or IT), and then used to predict **the HTER labels** for the test instances coming from the other domain (e.g. training on L, testing on IT).  **s175**: **This** is a strong evidence of the fact that, in case of domain changes, online models can still learn from new test instances even if they have a label distribution similar to the training set. | **A**: 1 |
| **B**: 2 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g16**: Focusing on the adaptability to user and domain changes, we report the results of comparative experiments with two online algorithms and the standard batch approach. The evaluation is carried out by measuring the global error of each algorithm on test sets featuring different degrees of similarity with the data used for training. Our results show that the sensitivity of online **QE** models to different distributions of training and test instances makes them more suitable than batch methods for integration in a **CAT** framework.  **g**76: In the last round of experiments we evaluate the reactivity of different online models to simultaneous user and domain changes. To this aim, our QE models are created using a training set coming from one domain (L or IT), and then used to predict **the HTER labels** for the test instances coming from the other domain ( e.g., training on L, testing on IT ). | **A**: 4  **B**: 3  **Number of dangling anaphors:** 3 |
| **Paragraph-composed summary** | **p12**: Focusing on the adaptability to user and domain changes, we report the results of comparative experiments with two online algorithms and the standard batch approach. The evaluation is carried out by measuring the global error of each algorithm on test sets featuring different degrees of similarity with the data used for training. Our results show that the sensitivity of online **QE** models to different distributions of training and test instances makes them more suitable than batch methods for integration in a **CAT** framework.  **p**64: In the last round of experiments we evaluate the reactivity of different online models to simultaneous user and domain changes. To this aim, our QE models are created using a training set coming from one domain (L or IT), and then used to predict **the HTER labels** for the test instances coming from the other domain ( e.g., training on L, testing on IT ). | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 3 |

Table E-6. Standard summary, produced summaries and manual readability annotations of P14-1070.

|  |  |  |
| --- | --- | --- |
| **P14-1070** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | In this paper, we investigate various strategies to predict both syntactic dependency parsing and contiguous multiword expression (MWE) recognition, testing them on the dependency version of French Treebank (Abeill´e and Barrier, 2004), as instantiated in the SPMRL Shared Task (Seddah et al., 2013). Our work focuses on using an alternative representation of syntactically regular MWEs, which captures their syntactic internal structure. We obtain a system with comparable performance to that of previous works on this dataset, but which predicts both syntactic dependencies and the internal structure of MWEs. This can be useful for capturing the various degrees of semantic compositionality of MWEs. |  |
| **Sentence-composed summary** | **s108**: REG-BY-PARSER: all regular MWE information (topology, status, POS) is predicted via dependency parsing, using representations with all information for regular **MWE**s encoded in topology and labels (**Figure 2**).  **s111**: training set: irregular MWEs merged into one token, regular MWEs are structured, and integration of regular MWE information into the labels (FCT\_r\_POS).  **s172**: The evaluation on “structured representation” can be interpreted as an evaluation of the parsing task plus the recognition of irregular MWEs only: both **LAS** and **UAS** are measured independently of errors on regular MWE status (note the UAS is exactly the same than in **the “labeled” case**). | **A**: 0 |
| **B**: 0 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g35**: IRREG-BY-PARSER: the MWE status, flat topology and POS are all predicted via dependency parsing, using representations for training and parsing, with all information for irregular **MWE**s encoded in topology and labels (as for in vain in **Figure 2**).  **g**38: REG-BY-PARSER: all regular MWE information (topology, status, POS) is predicted via dependency parsing, using representations with all information for regular MWEs encoded in topology and labels (Figure 2).  **g41**: parsing: (i) MWE analysis with classification of MWEs into regular or irregular, (ii) merge of predicted irregular MWEs, (iii) tagging and morphological prediction, (iv) parsing | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 2 |
| **Paragraph-composed summary** | **p26**: IRREG-BY-PARSER: the MWE status, flat topology and POS are all predicted via dependency parsing, using representations for training and parsing, with all information for irregular **MWE**s encoded in topology and labels (as for in vain in **Figure 2**).  **p**29: REG-BY-PARSER: all regular MWE information (topology, status, POS) is predicted via dependency parsing, using representations with all information for regular MWEs encoded in topology and labels (Figure 2).  **p32**: parsing: (i) MWE analysis with classification of MWEs into regular or irregular, (ii) merge of predicted irregular MWEs, (iii) tagging and morphological prediction, (iv) parsing | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 2 |

Table E-7. Standard summary, produced summaries and manual readability annotations of P14-1073.

|  |  |  |
| --- | --- | --- |
| **P14-1073** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Entity clustering must determine when two named-entity mentions refer to the same entity. Typical approaches use a pipeline architecture that clusters the mentions using fixed or learned measures of name and context similarity. In this paper, we propose a model for cross-document coreference resolution that achieves robustness by learning similarity from unlabeled data. The generative process assumes that each entity mention arises from copying and optionally mutating an earlier name from a similar context. Clustering the mentions into entities depends on recovering this copying tree jointly with estimating models of the mutation process and parent selection process. We present a block Gibbs sampler for posterior inference and an empirical evaluation on several datasets. |  |
| **Sentence-composed summary** | **s160**: Here p 0 ranges over **mentions** (including ♦) that precede x in **the ordering** i, and w(p 0 , x)—defined later in sec. 5.3.  **s177**: We first sample an ordering i♦ (the ordering of mentions with parent ♦, i.e. **all mentions**) uniformly at random from the set of orderings compatible with the current p.  **s214**: Given **the topics** z, the ordering i, and **the observed names**, we choose an x.p value according to its posterior probability.  **s243**: **That edge** explains a mention x as a mutation of some parent p in the context of a particular sample (ps, is, zs). | **A**: 0 |
| **B**: 0 |
| Number of dangling anaphors: 6 |
| **Group-composed summary** | **g48**: \todo[author=jason,color=RedOrange,fancyline,size=,]If we’re using [], then don’t we have to say how 𝒎 is sampled?  **g**102: The Ψx(x.z) factors in **(5)** approximate the topic model’s prior distribution over z. Ψx(x.z) is proportional to the probability that a Gibbs sampling step for an ordinary topic model would choose this value of x.z. This depends on whether—in the current sample—x.z is currently common in x’s document and x.t is commonly generated by x.z. It ignores the fact that we will also be resampling the topics of the other mentions. | **A**: 2 |
| **B**: 1 |
| Number of dangling anaphors: 1 |
| **Paragraph-composed summary** | **p50**: To model **this pragmatic effect**, we multiply our definition of Prθ(x.n | p.n) by an extra factor Pr(x.e | x) γ , where γ ≥ 0 is the effect strength.  **p**66: The Ψx(x.z) factors in **(5)** approximate the topic model’s prior distribution over z. Ψx(x.z) is proportional to the probability that a Gibbs sampling step for an ordinary topic model would choose this value of x.z. This depends on whether—in the current sample—x.z is currently common in x’s document and x.t is commonly generated by x.z. | **A**: 1 |
| **B**: 2 |
| Number of dangling anaphors: 2 |

Table E-8. Standard summary, produced summaries and manual readability annotations of P14-1077.

|  |  |  |
| --- | --- | --- |
| **P14-1077** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Most existing relation extraction models make predictions for each entity pair locally and individually, while ignoring implicit global clues available in the knowledge base, sometimes leading to conflicts among local predictions from different entity pairs. In this paper, we propose a joint inference framework that utilizes these global clues to resolve disagreements among local predictions. We exploit two kinds of clues to generate constraints which can capture the implicit type and cardinality requirements of a relation. Experimental results on three datasets, in both English and Chinese, show that our framework outperforms the state-of-theart relation extraction models when such clues are applicable to the datasets. And, we find that the clues learnt automatically from existing knowledge bases perform comparably to those refined by human. |  |
| **Sentence-composed summary** | **s45**: In order to implicitly capture the expected type and cardinality requirements for **a relation’s arguments**, we derive two kinds of clues from an existing KB, which are further utilized to discover the disagreements among **local candidate predictions**.  **s55**: For a candidate relation r ∈ Rt and a tuple t, we define Mr t as all t’s **mentions** whose candidate relations contain r.  **s100**: Given two relations r1 and r2, we query the KB for all tuples bearing the relation r1 or r2.  **s170**: However, in the Riedel’s dataset, Mintz++, the MaxEnt relation extractor, does not perform well, and **our framework** cannot improve its performance.  s209: We add clues according to their related relations’ proportions in the local predictions. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 4 |
| **Group-composed summary** | **g15**: Riedel et al. (2013) propose to use latent vectors to estimate the preferences between relations and entities. These can be considered as the latent type information of **the relations’ arguments**, which is learnt from various data sources. In contrast, our approach learn implicit clues from existing KBs, and jointly optimize **local predictions** among different entity tuples to capture both relation argument type clues and **cardinality clues**.  **g**88: In this paper, we make use of the global clues derived from KB to help resolve the disagreements among local relation predictions, thus reduce the incorrect predictions and improve the performance of relation extraction. Two kinds of clues, including implicit argument type information and argument cardinality information of relations are investigated. | **A**: 4 |
| **B**: 3 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p19**: **The previous categories of disagreements** are all based on the implicit type information of the relations’ arguments, Now we make use of the clues of argument cardinality requirements. Given a subject, some relations should have unique objects. For example, in Figure 1, given USA as the subject of the relation Capital, we can only accept one possible object, because there is great chance that a country only have one capital. On the other hand, given Washington D.C. as the object of the relation Capital, we can only accept one subject, since usually a city can only be the capital of one country or state. If these are violating in **the candidates**, we could know that there may be some incorrect predictions. | **A**: 4 |
| **B**: 3 |
| Number of dangling anaphors: 2 |

Table E-9. Standard summary, produced summaries and manual readability annotations of P14-1090.

|  |  |  |
| --- | --- | --- |
| **P14-1090** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Answering natural language questions using the Freebase knowledge base has recently been explored as a platform for advancing the state of the art in open domain semantic parsing. Those efforts map questions to sophisticated meaning representations that are then attempted to be matched against viable answer candidates in the knowledge base. Here we show that relatively modest information extraction techniques, when paired with a webscale corpus, can outperform these sophisticated approaches by roughly 34% relative gain. |  |
| **Sentence-composed summary** | **s45**: We call this a question feature graph, with every node and relation a potential feature for **this question**.  **s55**: Thus for the question, who is the father of King George VI, we ask two questions: does the mapping, 1. (coverage) contain the answer relation people.person.parents?  **s100**: 2. (precision) predict the answer relation from the question?  **s170**: For each question, we extracted all relations in its corresponding **topic graph**, and ranked each relation with whether it is the answer relation. | **A**: 3 |
| **B**: 4 |
| Number of dangling anaphors: 2 |
| **Group-composed summary** | **g125**: Also, we did an ablation test on **DEV** about how additional features on the mapping between Freebase relations and the original questions help, with three feature settings: 1) “basic” features include feature productions read off from **the feature graph** (Figure 1); 2) “+ word overlap” adds additional features on whether **sub-relations** have overlap with the question; and 3) “+ **CluewebMapping**” adds the ranking of relation prediction given the question according to CluewebMapping. | **A**: 2 |
| **B**: 3 |
| Number of dangling anaphors: 4 |
| **Paragraph-composed summary** | **p69**: Also, we did an ablation test on **DEV** about how additional features on the mapping between Freebase relations and the original questions help, with three feature settings: 1) “basic” features include feature productions read off from **the feature graph** (Figure 1); 2) “+ word overlap” adds additional features on whether **sub-relations** have overlap with the question; and 3) “+ **CluewebMapping**” adds the ranking of relation prediction given the question according to CluewebMapping. | **A**: 2 |
| **B**: 3 |
| **Number of dangling anaphors:** 4 |

Table E-10. Standard summary, produced summaries and manual readability annotations of P14-1091.

|  |  |  |
| --- | --- | --- |
| **P14-1091** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | A typical knowledge-based question answering (KB-QA) system faces two challenges: one is to transform natural language questions into their meaning representations (MRs); the other is to retrieve answers from knowledge bases (KBs) using generated MRs. Unlike previous methods which treat them in a cascaded manner, we present a translation-based approach to solve these two tasks in one unified framework. We translate questions to answers based on CYK parsing. Answers as translations of the span covered by each CYK cell are obtained by a question translation method, which first generates formal triple queries as MRs for the span based on question patterns and relation expressions, and then retrieves answers from a given KB based on triple queries generated. A linear model is defined over derivations, and minimum error rate training is used to tune feature weights based on a set of question-answer pairs. Compared to a KB-QA system using a state-of-the-art semantic parser, our method achieves better results. |  |
| **Sentence-composed summary** | **s35**: **The first half** (from Line 1 to Line 13) generates a **formal triple** set T for each unary **span** Q j i ∈ Q, using the question translation method QT rans(Q j i , KB) (Line 4), which takes Q j i as the input.  **s54**: Algorithm 2 shows how to generate formal triples for a span Q based on **question patterns** (QP-based question translation).  **s121**: hQPcount (·), which counts the number of triples in D that are generated by **QP-based question translation method**.  **s122**: • hREcount (·), which counts the number of triples in D that are generated by RE-based question translation method. | **A**: 0 |
| **B**: 1 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g17**: We first present our translation-based KB-QA method in Algorithm 1, which is used to generate **H(Q)** for each input **NL question Q**.  **g20**: **The first half** (from Line 1 to Line 13) generates a **formal triple** set T for each unary **span** Q j i ∈ Q, using the question translation method QT rans(Q j i , KB) (Line 4), which takes Q j i as the input.  **g64**: hQPcount (·), which counts the number of triples in D that are generated by **QP-based question translation method**.  **g65**: • hREcount (·), which counts the number of triples in D that are generated by RE-based question translation method. | **A**: 1 |
| **B**: 2 |
| Number of dangling anaphors: 6 |
| **Paragraph-composed summary** | **p13**: **The first half** (from Line 1 to Line 13) generates a **formal triple** set T for each unary **span** Q j i ∈ Q, using the question translation method QT rans(Q j i , KB) (Line 4), which takes Q j i as the input.  **p38**: hspan(·), which counts the number of spans in Q that are converted to formal triples. It controls the granularity of the spans used in question translation.  **p43**: hQPcount (·), which counts the number of triples in D that are generated by QP-based question translation method.. | **A**: 1  **B**: 2  **Number of dangling anaphors:** 3 |

Table E-11. Standard summary, produced summaries and manual readability annotations of P14-1113.

|  |  |  |
| --- | --- | --- |
| **P14-1113** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Semantic hierarchy construction aims to build structures of concepts linked by hypernym–hyponym (“is-a”) relations. A major challenge for this task is the automatic discovery of such relations. This paper proposes a novel and effective method for the construction of semantic hierarchies based on word embeddings, which can be used to measure the semantic relationship between words. We identify whether a candidate word pair has hypernym–hyponym relation by using the word-embedding-based semantic projections between words and their hypernyms. Our result, an F-score of 73.74%, outperforms the state-of-the-art methods on a manually labeled test dataset. Moreover, combining our method with a previous manually-built hierarchy extension method can further improve F-score to 80.29%. |  |
| **Sentence-composed summary** | **s11**: However, there usually also exists hypernym–hyponym relations among **these hypernyms**.  **s29**: Furthermore, we propose a piecewise linear projection method based on relation clustering to better model hypernym–hyponym relations (Section 3.3.2).  **s63**: In this paper, we aim to identify hypernym– hyponym relations using word embeddings, which have been shown to preserve good properties for capturing semantic relationship between words.  **s177**: We extract hypernym–hyponym relations in the Baidubaike corpus, which is also used to train word embeddings (Section 4.1).  **s224**: Our method based on word embeddings can discover more hypernym– hyponym relations than the previous methods can.  **s254**: Using the word embeddings, we learn the hypernym–hyponym relationship by estimating projection matrices which map words to their hypernyms. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 1 |
| **Group-composed summary** | **g20**: In this paper, we aim to identify hypernym– hyponym relations using word embeddings, which have been shown to preserve good properties for capturing semantic relationship between words.  **g22**: Then we elaborate on our proposed method composed of three major steps, namely, word embedding training, **projection learning**, and hypernym– hyponym relation identification.  **g67**: The training data for projection learning is collected from CilinE (Section 3.3.3). We obtain 15,247 word pairs of hypernym–hyponym relations (9,288 for direct relations and 5,959 for indirect relations).  **g101**: Some fine-grained relations exist in Wikipedia, but the coverage is limited. Our method based on word embeddings can discover more hypernym– hyponym relations than the previous methods can. | **A**: 4 |
| **B**: 3 |
| Number of dangling anaphors: 1 |
| **Paragraph-composed summary** | **p13**: The training data for **projection learning** is collected from CilinE (Section 3.3.3). We obtain 15,247 word pairs of hypernym–hyponym relations (9,288 for direct relations and 5,959 for indirect relations).  **p38**: We analyze error cases after experiments. Some cases are shown in **Figure 8**. We can see that there is only one general relation “植物 (plant)” −H→ “生物 (organism)” existing in CilinE. Some fine-grained relations exist in Wikipedia, but the coverage is limited. Our method based on word embeddings can discover more hypernym– hyponym relations than the previous methods can. When we combine the methods together, we get the correct hierarchy. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 2 |

Table E-12. Standard summary, produced summaries and manual readability annotations of P14-1128.

|  |  |  |
| --- | --- | --- |
| **P14-1128** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | This study investigates on building a better Chinese word segmentation model for statistical machine translation. It aims at leveraging word boundary information, automatically learned by bilingual character-based alignments, to induce a preferable segmentation model. We propose dealing with the induced word boundaries as soft constraints to bias the continuous learning of a supervised CRFs model, trained by the treebank data (labeled), on the bilingual data (unlabeled). The induced word boundary information is encoded as a graph propagation constraint. The constrained model induction is accomplished by using posterior regularization algorithm. The experiments on a Chinese-to-English machine translation task reveal that the proposed model can bring positive segmentation effects to translation quality. |  |
| **Sentence-composed summary** | **s13**: We propose leveraging **the bilingual knowledge** to form learning constraints that guide a supervised segmentation model toward a better solution for **SMT**.  **s43**: Rather than playing the “hard” uses of the bilingual segmentation knowledge, i.e., directly merging “char-to-word” alignments to words as supervisions, this study extracts word boundary information of characters from the alignments as soft constraints to regularize a CRFs model’s learning.  **s54**: One of our main objectives is to bias CRFs model’s learning on **unlabeled data**, under a non-linear **GP** constraint encoding the bilingual knowledge.  **s189**: The type-level word boundary distributions, induced by the character-based alignment (VES-NOGP), and **the graph propagation** (VES-GPPL), are regarded as virtual evidences to bias CRFs model’s learning on the unlabeled data. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g2**: This paper proposes an alternative Chinese Word Segmentation (CWS) model adapted to the **SMT** task, which seeks not only to maintain the advantages of a monolingual supervised model, having hand-annotated linguistic knowledge, but also to assimilate the relevant bilingual segmentation nature. We propose leveraging the bilingual knowledge to form learning constraints that guide a supervised segmentation model toward a better solution for SMT. Besides the bilingual motivated models, character-based alignment is also employed to achieve the mappings of the successive Chinese characters and the target language words.  **g71**: The type-level word boundary distributions, induced by the character-based alignment (VES-NOGP), and **the graph propagation** (VES-GPPL), are regarded as virtual evidences to bias **CRFs model**’s learning on the **unlabeled data**. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 4 |
| **Paragraph-composed summary** | **p2**: This paper proposes an alternative Chinese Word Segmentation (CWS) model adapted to the **SMT** task, which seeks not only to maintain the advantages of a monolingual supervised model, having hand-annotated linguistic knowledge, but also to assimilate the relevant bilingual segmentation nature. We propose leveraging the bilingual knowledge to form learning constraints that guide a supervised segmentation model toward a better solution for SMT. Besides the bilingual motivated models, character-based alignment is also employed to achieve the mappings of the successive Chinese characters and the target language words. Instead of directly merging the characters into concrete segmentations, this work attempts to extract word boundary distributions for characterlevel trigrams (types) from the “chars-to-word” mappings. | **A**: 5 |
| **B**: 5 |
| Number of dangling anaphors: 1 |

Table E-13. Standard summary, produced summaries and manual readability annotations of P14-1132.

|  |  |  |
| --- | --- | --- |
| **P14-1132** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Following up on recent work on establishing a mapping between vector-based semantic embeddings of words and the visual representations of the corresponding objects from natural images, we first present a simple approach to cross-modal vector-based semantics for the task of zero-shot learning, in which an image of a previously unseen object is mapped to a linguistic representation denoting its word. We then introduce fast mapping, a challenging and more cognitively plausible variant of the zero-shot task, in which the learner is exposed to new objects and the corresponding words in very limited linguistic contexts. By combining prior linguistic and visual knowledge acquired about words and their objects, as well as exploiting the limited new evidence available, the learner must learn to associate new objects with words. Our results on this task pave the way to realistic simulations of how children or robots could use existing knowledge to bootstrap grounded semantic knowledge about new concepts. |  |
| **Sentence-composed summary** | **s21**: **In this setting**, we assume that our system possesses linguistic and visual information for a set of concepts in the form of text-based representations of words and image-based vectors of the corresponding objects, used for vision-to-language-mapping training.  **s53**: Frome et al. (2013) use linear regression to transform vector-based image representations onto vectors representing the same concepts in linguistic semantic space.  **s66**: Objects corresponding to concepts are represented in **visual terms** by vectors in an image-based semantic space (Section 4.2).  **s111**: The process of learning to map objects to the their word label is implemented by training a projection function fprojv→w from the visual onto the linguistic semantic space.  **s112**: For the learning, we use a set of Ns **seen concepts** for which we have both image-based visual representations Vs ∈ R Ns×dv and text-based linguistic representations Ws ∈ R Ns×dw. | **A**: 1 |
| **B**: 0 |
| Number of dangling anaphors: 3 |
| **Group-composed summary** | **g9**: We first test the effectiveness of our **cross-modal semantic space** on the so-called zero-shot learning task (Palatucci et al., 2009), which has recently been explored in the machine learning community (Frome et al., 2013; Socher et al., 2013)..  **g28**: Concretely, we assume **that concepts**, denoted for convenience by word labels, are represented in linguistic terms by vectors in a text-based distributional semantic space (see Section 4.3). Objects corresponding to concepts are represented in visual terms by vectors in an image-based semantic space (Section 4.2). For a subset of concepts (e.g., a set of animals, a set of vehicles), we possess information related to both their linguistic and visual representations. During training, this cross-modal vocabulary is used to induce a projection function (Section 4.4), which – intuitively – represents a mapping between visual and linguistic dimensions. Thus, this function, given a visual vector, returns its corresponding linguistic representation. At test time, the system is presented with a previously unseen object (e.g., wampimuk). | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 2 |
| **Paragraph-composed summary** | **p11**: Concretely, we assume **that concepts**, denoted for convenience by word labels, are represented in linguistic terms by vectors in a text-based distributional semantic space (see Section 4.3). Objects corresponding to concepts are represented in visual terms by vectors in an image-based semantic space (Section 4.2). For a subset of concepts (e.g., a set of animals, a set of vehicles), we possess information related to both their linguistic and visual representations. During training, this cross-modal vocabulary is used to induce a projection function (Section 4.4), which – intuitively – represents a mapping between visual and linguistic dimensions. Thus, this function, given a visual vector, returns its corresponding linguistic representation. At test time, the system is presented with a previously unseen object (e.g., wampimuk). This object is projected onto the linguistic space and associated with the word label of the nearest neighbor in that space (degus in Figure 1b). | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 1 |

Table E-14. Standard summary, produced summaries and manual readability annotations of P14-1133.

|  |  |  |
| --- | --- | --- |
| **P14-1133** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | A central challenge in semantic parsing is handling the myriad ways in which knowledge base predicates can be expressed. Traditionally, semantic parsers are trained primarily from text paired with knowledge base information. Our goal is to exploit the much larger amounts of raw text not tied to any knowledge base. In this paper, we turn semantic parsing on its head. Given an input utterance, we first use a simple method to deterministically generate a set of candidate logical forms with a canonical realization in natural language for each. Then, we use a paraphrase model to choose the realization that best paraphrases the input, and output the corresponding logical form. We present two simple paraphrase models, an association model and a vector space model, and train them jointly from question-answer pairs. Our system PARASEMPRE improves stateof-the-art accuracies on two recently released question-answering datasets. |  |
| **Sentence-composed summary** | **s9**: Given an input utterance, we first use a simple deterministic procedure to construct a manageable set of candidate **logical forms** (ideally, we would generate **canonical utterances** for all possible logical forms, but this is intractable).  **s41**: Given an utterance x and the KB, we construct a set of candidate logical forms Zx, and then for each z ∈ Zx generate a small set of canonical natural language utterances Cz.  **s51**: Note that the candidate set of logical forms Zx and canonical utterances Cx are constructed during **the canonical utterance construction phase**.  **s71**: Given an input utterance x, we first construct a set of logical forms Zx, and then generate canonical utterances from each z ∈ Zx. | **A**: 3 |
| **B**: 2 |
| Number of dangling anaphors: 3 |
| **Group-composed summary** | **g2**: In this paper, we present a novel approach for semantic parsing based on paraphrasing that can exploit large amounts of text not covered by the KB (Figure 1). Our approach targets factoid questions with a modest amount of compositionality. Given an input utterance, we first use a simple deterministic procedure to construct a manageable set of candidate **logical forms** (ideally, we would generate **canonical utterances** for all possible logical forms, but this is intractable). Next, we heuristically generate canonical utterances for each logical form based on the text descriptions of predicates from the KB. Finally, we choose the canonical utterance that best paraphrases the input utterance, and thereby the logical form that generated it. We use two complementary paraphrase models: an association model based on aligned phrase pairs extracted from a monolingual parallel corpus, and a vector space model, which represents each utterance as a vector and learns a similarity score between them. | **A**: 5 |
| **B**: 5 |
| Number of dangling anaphors: 2 |
| **Paragraph-composed summary** | **p2**: In this paper, we present a novel approach for semantic parsing based on paraphrasing that can exploit large amounts of text not covered by the KB (Figure 1). Our approach targets factoid questions with a modest amount of compositionality. Given an input utterance, we first use a simple deterministic procedure to construct a manageable set of candidate **logical forms** (ideally, we would generate **canonical utterances** for all possible logical forms, but this is intractable). Next, we heuristically generate canonical utterances for each logical form based on the text descriptions of predicates from the KB. Finally, we choose the canonical utterance that best paraphrases the input utterance, and thereby the logical form that generated it. We use two complementary paraphrase models: an association model based on aligned phrase pairs extracted from a monolingual parallel corpus, and a vector space model, which represents each utterance as a vector and learns a similarity score between them. | **A**: 5 |
| **B**: 5 |
| Number of dangling anaphors: 2 |

Table E-15. Standard summary, produced summaries and manual readability annotations of P14-1136.

|  |  |  |
| --- | --- | --- |
| **P14-1136** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | We present a novel technique for semantic frame identification using distributed representations of predicates and their syntactic context; this technique leverages automatic syntactic parses and a generic set of word embeddings. Given labeled data annotated with frame-semantic parses, we learn a model that projects the set of word representations for the syntactic context around a predicate to a low dimensional representation. The latter is used for semantic frame identification; with a standard argument identification method inspired by prior work, we achieve state-of-the-art results on FrameNet-style frame-semantic analysis. Additionally, we report strong results on PropBank-style semantic role labeling in comparison to prior work. |  |
| **Sentence-composed summary** | **s9**: Given a dependency parse (1) **the model** extracts all words matching a set of paths from **the frame** **evoking predicate** and its direct dependents (2).  **s41**: We learn the initial embedding representations for our frame identification model (§3) using a deep neural language model similar to the one proposed by Bengio et al. (2003).  **s51**: The WSABIE EMBEDDING model from **§3** performs significantly better than the LOG-LINEAR WORDS baseline, while LOG-LINEAR EMBEDDING underperforms in every metric.  **s71**: Consequently, the WSABIE EMBEDDING model can share more information between different examples in the training data than the LOG-LINEAR EMBEDDING model. | **A**: 2 |
| **B**: 3 |
| Number of dangling anaphors: 4 |
| **Group-composed summary** | **g2**: Frame Lexicon In our experimental setup, we scanned the XML files in the “frames” directory of the FrameNet 1.5 release, which lists all the frames, the corresponding roles and the associated lexical units, and created a frame lexicon to be used in our frame and argument identification models.  **g2**: **Table 2** presents accuracy results on frame identification. We do not report partial frame accuracy that has been reported by **prior work**. We present results on all predicates, **ambiguous predicates** seen in the lexicon or the training data, and rare ambiguous predicates that appear ≤ 11 times in the training data. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p2**: **Table 2** presents accuracy results on frame identification. We do not report partial **frame** accuracy that has been reported by **prior work**. We present results on all predicates, **ambiguous predicates** seen in **the lexicon** or the training data, and rare ambiguous predicates that appear ≤ 11 times in the training data. The WSABIE EMBEDDING model from **§3** performs significantly better than the LOG-LINEAR WORDS baseline, while LOG-LINEAR EMBEDDING underperforms in every metric. For **the SEMAFOR LEXICON setup**, we also compare with the state of the art from Das et al. (2014), who used a semi-supervised learning method to improve upon a supervised latent-variable log-linear model. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 7 |

Table E-16. Standard summary, produced summaries and manual readability annotations of P14-1138.

|  |  |  |
| --- | --- | --- |
| **P14-1138** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | This study proposes a word alignment model based on a recurrent neural network (RNN), in which an unlimited alignment history is represented by recurrently connected hidden layers. We perform unsupervised learning using noise-contrastive estimation (Gutmann and Hyvarinen, 2010; Mnih and Teh, ¨ 2012), which utilizes artificially generated negative samples. Our alignment model is directional, similar to the generative IBM models (Brown et al., 1993). To overcome this limitation, we encourage agreement between the two directional models by introducing a penalty function that ensures word embedding consistency across two directional models during training. The RNN-based model outperforms the feed-forward neural network-based model (Yang et al., 2013) as well as the IBM Model 4 under Japanese-English and French-English word alignment tasks, and achieves comparable translation performance to those baselines for Japanese-English and Chinese-English translation tasks. |  |
| **Sentence-composed summary** | **s36**: The IBM Models 1 and 2 and the HMM model decompose **it** into an **alignment probability** pa and a **lexical translation probability** pt as.  **s160**: We evaluated the proposed RNN-based alignment models against two baselines: the IBM Model 4 and the **FFNN**-based model with one hidden layer.  **s207**: In **NTCIR** and **FBIS**, each alignment model was trained from the randomly sampled 100 K data, and then a translation model was trained from all the training data that was word-aligned by the alignment model. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 6 |
| **Group-composed summary** | **g16**: The IBM Models 1 and 2 and the HMM model decompose **it** into an **alignment probability** pa and a **lexical translation probability** pt as.  **g76**: We evaluated the proposed RNN-based alignment models against two baselines: the IBM Model 4 and the **FFNN**-based model with one hidden layer.  **g103**: **Table 4** shows the alignment performance on BT EC with various training data sizes, i.e., training data for **IWSLT** (40 K), training data for **BTEC** (9 K), and the randomly sampled 1 K data from the BTEC training data. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 7 |
| **Paragraph-composed summary** | **p42**: We evaluated the proposed RNN-based alignment models against two baselines: the IBM Model 4 and the **FFNN**-based model with one hidden layer. The IBM Model 4 was trained by previously presented **model sequence schemes** (Och and Ney, 2003).  **p54**: **Table 4** shows the alignment performance on BT EC with various training data sizes, i.e., training data for **IWSLT** (40 K), training data for **BTEC** (9 K), and the randomly sampled 1 K data from the BTEC training data. Note that RNNs+c(R) cannot be trained from the 40 K data because the 40 K data does not have gold standard word alignments. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 5 |

Table E-17. Standard summary, produced summaries and manual readability annotations of P14-1140.

|  |  |  |
| --- | --- | --- |
| **P14-1140** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | In this paper, we propose a novel recursive recurrent neural network (R2NN) to model the end-to-end decoding process for statistical machine translation. R2NN is a combination of recursive neural network and recurrent neural network, and in turn integrates their respective capabilities: (1) new information can be used to generate the next hidden state, like recurrent neural networks, so that language model and translation model can be integrated naturally; (2) a tree structure can be built, as recursive neural networks, so as to generate the translation candidates in a bottom up manner. A semi-supervised training approach is proposed to train the parameters, and the phrase pair embedding is explored to model translation confidence directly. Experiments on a Chinese to English translation task show that our proposed R2NN can outperform the state-of-the-art baseline by about 1.5 points in BLEU. |  |
| **Sentence-composed summary** | **s15**: Auli et al. (2013) propose a joint language and translation model, based on a recurrent neural network.  **s19**: Word embedding is used as the input to learn **translation confidence score**, which is combined with commonly used features in **the conventional log-linear model**.  **s29**: So as to model the translation confidence for a translation phrase pair, we initialize the phrase pair embedding by leveraging **the sparse features** and recurrent neural network.  **s107**: We also explore phrase pair embedding method to model translation confidence directly, which is introduced in Section 5.  **s152**: **In this section**, we split the phrase pair embedding into two parts to model the translation confidence directly: translation confidence with sparse features and translation confidence with recurrent neural network.  **s153**: We first get two **translation confidence vectors** separately using sparse features and recurrent neural network, and then concatenate them to be the phrase pair embedding. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g46**: Liu et al. (2013) apply DNN to **SMT** decoding, but not in a recursive manner. A feature is learnt via a one-hidden-layer neural network, and the embedding of words in the phrase pairs are used as the input vector. Our model generates the representation of a translation pair based on its **child nodes**.  **g47**: Li et al. (2013) also generate the representation of phrase pairs in a recursive way. In their work, the representation is optimized to learn a distortion model using recursive neural network, only based on the representation of the child nodes. Our R2NN is used to model the end-to-end translation process, with **recurrent global information** added. We also explore phrase pair embedding method to model translation confidence directly, which is introduced in Section 5.  **g89**: In this paper, we propose a Recursive Recurrent Neural Network(R2NN) to combine the recurrent neural network and recursive neural network. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p25**: Liu et al. (2013) apply DNN to **SMT** decoding, but not in a recursive manner. A feature is learnt via a one-hidden-layer neural network, and the embedding of words in the phrase pairs are used as the input vector. Our model generates the representation of a translation pair based on its **child nodes**. Li et al. (2013) also generate the representation of phrase pairs in a recursive way. In their work, the representation is optimized to learn a distortion model using recursive neural network, only based on the representation of the child nodes. Our R2NN is used to model the end-to-end translation process, with **recurrent global information** added. We also explore phrase pair embedding method to model translation confidence directly, which is introduced in Section 5.  **p59**: In this paper, we propose a Recursive Recurrent Neural Network(R2NN) to combine the recurrent neural network and recursive neural network. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 3 |

Table E-18 Standard summary, produced summaries and manual readability annotations of P14-1144

|  |  |  |
| --- | --- | --- |
| **P14-1144** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Recently, researchers have begun exploring methods of scoring student essays with respect to particular dimensions of quality such as coherence, technical errors, and prompt adherence. The work on modeling prompt adherence, however, has been focused mainly on whether individual sentences adhere to the prompt. We present a new annotated corpus of essay-level prompt adherence scores and propose a feature-rich approach to scoring essays along the prompt adherence dimension. Our approach significantly outperforms a knowledge-lean baseline prompt adherence scoring system yielding improvements of up to 16.6%. |  |
| **Sentence-composed summary** | **s91**: To obtain feature values of **the first type**, we take the **RI** similarities between the whole essay and each set of **prompt adherence keywords** from **the prompt’s components**.  **s98**: **These topics** should not diminish the essay’s prompt adherence score because they are at least related to **prompt concepts**.  **s110**: Since **the latter topic** is discussed so much in **the essay** and does not appear to have much to do with the military prompt, this essay should probably get a bad prompt adherence score. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 8 |
| **Group-composed summary** | **g36**: We form two types of features from **prompt adherence keywords**. While both types of features measure how much each **prompt component** was discussed in an essay, they differ in how they encode the information. To obtain feature values of the first type, we take the **RI** similarities between the whole essay and each set of prompt adherence keywords from the prompt’s components. This results in one to three features, as some prompts have one component while others have up to three. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p26**: We form two types of features from **prompt adherence keywords**. While both types of features measure how much each **prompt component** was discussed in an essay, they differ in how they encode the information. To obtain feature values of the first type, we take the **RI** similarities between the whole essay and each set of prompt adherence keywords from the prompt’s components. This results in one to three features, as some prompts have one component while others have up to three. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 3 |

Table E-19 Standard summary, produced summaries and manual readability annotations of P14-2008

|  |  |  |
| --- | --- | --- |
| **P14-2008** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Recent work classifying citations in scientific literature has shown that it is possible to improve classification results with extensive feature engineering. While this result confirms that citation classification is feasible, there are two drawbacks to this approach: (i) it requires a large annotated corpus for supervised classification, which in the case of scientific literature is quite expensive; and (ii) feature engineering that is too specific to one area of scientific literature may not be portable to other domains, even within scientific literature. In this paper we address these two drawbacks. First, we frame citation classification as a domain adaptation task and leverage the abundant labeled data available in other domains. Then, to avoid over-engineering specific citation features for a particular scientific domain, we explore a deep learning neural network approach that has shown to generalize well across domains using unigram and bigram features. We achieve better citation classification results with this cross-domain approach than using in-domain classification. |  |
| **Sentence-composed summary** | **s14**: We treat citation polarity classification as a sentiment analysis domain adaptation task and therefore must be careful not to define features that are too domain specific.  **s24**: We are interested in domain adaptation for citation classification and therefore need a target dataset of citations and a **non-citation source dataset**.  **s80**: **These experiments** should help answer two questions does a larger amount of training data, even if out of domain, improve citation classification; and how well do the different **product domains** generalize to citations (i.e.,, which domains are most similar to citations).  **s83**: Our initial results show that using **mSDA** for domain adaptation to citations actually outperforms in-domain classification.  **s88**: We can see that using a larger dataset, even if out of domain, does improve citation classification.  **s90**: Using a larger training set, along with mSDA, which makes use of **the unlabeled data**, leads to the best results for citation classification.  **s128**: **This semi-supervised domain adaptation approach** outperforms the in-domain citation polarity classification and other fully supervised domain adaptation approaches. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 6 |
| **Group-composed summary** | **g30**: Our initial experiments simply extend those of Chen et al. (2012) (and others who have used **MDSD**) by adding another domain, citations. We train on each of the domains from the MDSD –books, dvd, electronics, and kitchen – and test on the citation data. We split the labeled data 80/20 following Blitzer et al. (2007) (cf. Chen et al. (2012) train on all “labeled” data and test on the “unlabeled” data). These experiments should help answer two questions: does a larger amount of training data, even if out of domain, improve **citation classification**; and how well do the different **product domains** generalize to citations (i.e., which domains are most similar to citations)?  **g31**: In contrast to previous work using MDSD, a lot of the work in domain adaptation also leverages a small amount of labeled **target data**. In our second set of experiments, we follow the domain adaptation approaches described in (Daume´ III, 2007) and train on **product review** and **citation data** before testing on citations. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 6 |
| **Paragraph-composed summary** | **p13**: Our initial experiments simply extend those of Chen et al. (2012) (and others who have used **MDSD**) by adding another domain, citations. We train on each of the domains from the MDSD –books, dvd, electronics, and kitchen – and test on the citation data. We split the labeled data 80/20 following Blitzer et al. (2007) (cf. Chen et al. (2012) train on all “labeled” data and test on the “unlabeled” data). These experiments should help answer two questions: does a larger amount of training data, even if out of domain, improve **citation classification**; and how well do the different **product domains** generalize to citations (i.e., which domains are most similar to citations)?  **p15**: Our initial results show that using **mSDA** for domain adaptation to citations actually outperforms in-domain classification. In **Figure 1** we compare citation classification with mSDA to the SVM baseline. | **A**: 4 |
| **B**: 4 |
| **Number of dangling anaphors:** 5 |

Table E-20 Standard summary, produced summaries and manual readability annotations of P14-2010

|  |  |  |
| --- | --- | --- |
| **P14-2010** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Supervised text classification algorithms require a large number of documents labeled by humans that involve a labor-intensive and time consuming process. In this paper, we propose a weakly supervised algorithm in which supervision comes in the form of labeling of Latent Dirichlet Allocation (LDA) topics. We then use this weak supervision to “sprinkle” artificial words to the training documents to identify topics in accordance with the underlying class structure of the corpus based on the higher order word associations. We evaluate this approach to improve performance of text classification on three real world datasets. |  |
| **Sentence-composed summary** | **s6**: (Blei et al., 2003) used LDA topics as features in text classification, but they use **labeled documents** while learning a classifier sLDA (Blei and McAuliffe, 2007) , DiscLDA (Lacoste-Julien et al., 2008) and MedLDA (Zhu et al., 2009) are few extensions of LDA which model both class labels and words in the documents.  **s9**: **In this approach**, a topic model on a given set of unlabeled training documents is constructed using LDA, then an annotator assigns a class label to some topics based on their most probable words.  **s18**: As in **ClassifyLDA,** we ask an annotator to assign class labels to a set of topics inferred on the unlabeled training documents. | **A**: 1 |
| **B**: 2 |
| Number of dangling anaphors: 3 |
| **Group-composed summary** | **g28**: In our text classification algorithm, we first infer a set of topics on the given unlabeled document corpus. We then ask a human annotator to assign one or more class labels to the topics based on their most probable words. We use these labeled topics to create a new LDA model **as follows**. If the topic assigned to the word w at the position n in document d is t, then we replace it by the class label assigned to the topic t. | **A**: 4 |
| **B**: 5 |
| Number of dangling anaphors: 1 |
| **Paragraph-composed summary** | **p23**: In our text classification algorithm, we first infer a set of topics on the given unlabeled document corpus. We then ask a human annotator to assign one or more class labels to the topics based on their most probable words. We use these labeled topics to create a new LDA model **as follows**. If the topic assigned to the word w at the position n in document d is t, then we replace it by the class label assigned to the topic t. | **A**: 4 |
| **B**: 5 |
| Number of dangling anaphors: 1 |

Table E-21 Standard summary, produced summaries and manual readability annotations of P14-2036

|  |  |  |
| --- | --- | --- |
| **P14-2036** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | In this paper, we propose a novel model for enriching the content of microblogs by exploiting external knowledge, thus improving the data sparseness problem in short text classification. We assume that microblogs share the same topics with external knowledge. We first build an optimization model to infer the topics of microblogs by employing the topic-word distribution of the external knowledge. Then the content of microblogs is further enriched by relevant words from external knowledge. Experiments on microblog classification show that our approach is effective and outperforms traditional text classification methods. |  |
| **Sentence-composed summary** | **s6**: In LDA, each document has a distribution over all topics P(zk|dj ), and each topic has a distribution over all words P(wi |zk), where zk, dj and wi represent the topic, document and word respectively.  **s18**: Differing from **step (a)**, the method used for topic inference for microblogs is not directly running LDA estimation on microblog collection but following the topics from **external knowledge** to ensure topic consistence. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 2 |
| **Group-composed summary** | **g28**: Motivated by the idea of using topic model and external knowledge mentioned above, we present an LDA-based enriching method using the news corpus, and apply it to the task of microblog classification. The basic assumption in our model is that news articles and microblogs tend to share the same topics. We first infer the topic distribution of each microblog based on the topic-word distribution of news corpus obtained by the LDA estimation. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 0 |
| **Paragraph-composed summary** | **p23**: Motivated by the idea of using topic model and external knowledge mentioned above, we present an LDA-based enriching method using the news corpus, and apply it to the task of microblog classification. The basic assumption in our model is that news articles and microblogs tend to share the same topics. We first infer the topic distribution of each microblog based on the topic-word distribution of news corpus obtained by the LDA estimation. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 0 |

Table E-22 Standard summary, produced summaries and manual readability annotations of P14-2045

|  |  |  |
| --- | --- | --- |
| **P14-2045** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | A common task in qualitative data analysis is to characterize the usage of a linguistic entity by issuing queries over syntactic relations between words. Previous interfaces for searching over syntactic structures require programming-style queries. User interface research suggests that it is easier to recognize a pattern than to compose it from scratch; therefore, interfaces for non-experts should show previews of syntactic relations. What these previews should look like is an open question that we explored with a 400-participant Mechanical Turk experiment. We found that syntactic relations are recognized with 34% higher accuracy when contextual examples are shown than a baseline of naming the relations alone. This suggests that user interfaces should display contextual examples of syntactic relations to help users choose between different relations. |  |
| **Sentence-composed summary** | **s27**: Our results confirm that showing **examples** in the form of words or phrases significantly improves the accuracy with which grammatical relationships are recognized over the standard baseline of showing the relation name with blanks.  **s31**: In **each task**, **they** were shown a list of sentences containing a particular syntactic relationship between highlighted words.  **s38**: Grammatical relations are identified more accurately when shown with examples of **contextualizing words or phrases** than without.  **s18**: Participants in conditions that showed examples (phrases and words) were significantly more accurate at identifying the relations than participants in the baseline condition.  **s89**: Phrases significantly outperformed words and baseline for **clausal relations**.  **s104**: A list of phrases is the most recognizable presentation for clausal relationships (34% better than the baseline), and is as good as a list of words for the other types of relations, except **adverb modifiers**. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 6 |
| **Group-composed summary** | **g8**: Our results confirm that showing **examples** in the form of words or phrases significantly improves the accuracy with which grammatical relationships are recognized over the standard baseline of showing the relation name with blanks.  **g53**: The results (**Figure 2**) confirm our hypothesis. Participants in conditions that showed examples (phrases and words) were significantly more accurate at identifying the relations than participants in the baseline condition.  **g59**: The results imply that user interfaces for syntactic search should show candidate relationships augmented with a list of phrases in which they occur. A list of phrases is the most recognizable presentation for **clausal relationships** (34% better than the baseline), and is as good as a list of words for the other types of relations, except **adverb modifiers**. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 4 |
| **Paragraph-composed summary** | **p12**: To test **it**, participants were given a series of identification tasks. In each task, they were shown a list of 8 sentences, each containing a particular relationship between highlighted words. They were asked to identify the relationship from a list of 4 choices.  **p42**: **The results** imply that user interfaces for syntactic search should show candidate relationships augmented with a list of phrases in which they occur. A list of phrases is the most recognizable presentation for **clausal relationships** (34% better than **the baseline**), and is as good as a list of words for the other types of relations, except adverb modifiers. For adverb modifiers, the list of words is the most recognizable presentation. This is likely because English adverbs usually end in ‘-ly’ are therefore a distinctive set of words. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 4 |

Table E-23 Standard summary, produced summaries and manual readability annotations of P14-2047

|  |  |  |
| --- | --- | --- |
| **P14-2047** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | We present a study of aspects of discourse structure — specifically discourse devices used to organize information in a sentence — that significantly impact the quality of machine translation. Our analysis is based on manual evaluations of translations of news from Chinese and Arabic to English. We find that there is a particularly strong mismatch in the notion of what constitutes a sentence in Chinese and English, which occurs often and is associated with significant degradation in translation quality. Also related to lower translation quality is the need to employ multiple explicit discourse connectives (because, but, etc.), as well as the presence of ambiguous discourse connectives in the English translation. Furthermore, the mismatches between discourse expressions across languages significantly impact translation quality. |  |
| **Sentence-composed summary** | **s13**: For Chinese, there are 305 sentences (segments) translated to English by three different translation systems.  **s42**: **The test** revealed that there is a significant difference in translation quality between **1-1 and 1-many segments** for Chinese but not for Arabic.  **s55**: We compare the translation quality obtained on segments with **reference translation** containing no **discourse connective**, exactly one discourse connective and more than one discourse connective.  **s64**: Here we compare the translation quality of segments which contain ambiguous discourse connectives in the reference translation to those that do not.  **s73**: The finding that **discourse connective ambiguity** is associated with change in translation quality for Chinese but not for Arabic is rather interesting.  **s105**: None of **these discourse factors** has a significant impact on translation quality from Arabic to English. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 6 |
| **Group-composed summary** | **g50**: We showed that translation from Chinese to English is made more difficult by various discourse events such as the use of discourse connectives, the ambiguity of the connectives and the type of relations they signal. None of these discourse factors has a significant impact on translation quality from Arabic to English. Translation quality from both languages is adversely affected by translations of discourse relations expressed implicitly in one language but explicitly in the other or by **paired connectives**. Our experiments indicate that discourse usage may affect machine translation between some language pairs but not others, and for particular relations such as **CONTINGENCY**. Finally, we established the need to identify sentences in the source language that would be translated into multiple sentences in English. | **A**: 5 |
| **B**: 4 |
| Number of dangling anaphors: 2 |
| **Paragraph-composed summary** | **p22**: A number of discourse connectives are ambiguous with respect to the discourse relation they convey. For example, while can signal either COMPARISON or TEMPORAL relations and since can signal either CONTINGENCY or TEMPORAL. In translation this becomes a problem when the ambiguity is present in one language but not in the other. In such cases the sense in source ought to be disambiguated before translation. Here we compare the translation quality of segments which contain ambiguous discourse connectives in the **reference translation** to those that do not. This analysis gives lower bound on the translation quality degradation associated with discourse phenomena as it does not capture problems arising from connective ambiguity on the source side. | **A**: 5 |
| **B**: 4 |
| Number of dangling anaphors: 1 |

Table E-24 Standard summary, produced summaries and manual readability annotations of P14-2053

|  |  |  |
| --- | --- | --- |
| **P14-2053** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | As students read expository text, comprehension is improved by pausing to answer questions that reinforce the material. We describe an automatic question generator that uses semantic pattern recognition to create questions of varying depth and type for self-study or tutoring. Throughout, we explore how linguistic considerations inform system design. In the described system, semantic role labels of source sentences are used in a domain-independent manner to generate both questions and answers related to the source sentence. Evaluation results show a 44% reduction in the error rate relative to the best prior systems, averaging over all metrics, and up to 61% reduction in the error rate on grammaticality judgments. |  |
| **Sentence-composed summary** | **s17**: Lindberg et al. (2013), which used semantic role labeling to identify **patterns** in the source text from which questions can be generated.  **s18**: This work most closely parallels our own with a few exceptions: our system only asks questions that can be answered from the source text, our approach is domain-independent, and the patterns also identify the answer to the question.  **s25**: Generation patterns specify the text, verb forms and **semantic arguments** from the source sentence to form the question.  **s55**: The pattern also filters out sentences with **A0 or A2**. The patterns are designed to match only the arguments used as part of the question or the answer, in order to prevent over generation of questions.  **s65**: For example, **this sentence** could also match patterns to generate questions **such as**: | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g7**: Lindberg et al. (2013), which used semantic role labeling to identify **patterns** in the source text from which questions can be generated. This work most closely parallels our own with a few exceptions: our system only asks questions that can be answered from the source text, our approach is domain-independent, and the patterns also identify the answer to the question.  **g26**: **Table 2** provides examples of generated questions. **The pattern** that generated Question 1 requires argument A1 (underlined in Table 2) and **a causation ArgM** (italicized). The pattern also filters out sentences with **A0 or A2**. The patterns are designed to match only the arguments used as part of the question or the answer, in order to prevent over generation of questions. | **A**: 2 |
| **B**: 1 |
| Number of dangling anaphors: 5 |
| **Paragraph-composed summary** | **p3**: Another recent approach is Lindberg et al. (2013), which used semantic role labeling to identify **patterns** in the source text from which questions can be generated.  **p22**: **Table 2** provides examples of generated questions. **The pattern** that generated Question 1 requires argument A1 (underlined in Table 2) and **a causation ArgM** (italicized). The pattern also filters out sentences with **A0 or A2**. The patterns are designed to match only the arguments used as part of the question or the answer, in order to prevent over generation of questions. **The system** inserted the correct forms of release and do, and ignored the phrase As this occurs since it is not part of the **semantic argument**. | **A**: 1 |
| **B**: 2 |
| Number of dangling anaphors: 7 |

Table E-25 Standard summary, produced summaries and manual readability annotations of P14-2068

|  |  |  |
| --- | --- | --- |
| **P14-2068** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | How do journalists mark quoted content as certain or uncertain, and how do readers interpret these signals? Predicates such as thinks, claims, and admits offer a range of options for framing quoted content according to the author’s own perceptions of its credibility. We gather a new dataset of direct and indirect quotes from Twitter, and obtain annotations of the perceived certainty of the quoted statements. We then compare the ability of linguistic and extra-linguistic features to predict readers’ assessment of the certainty of quoted content. We see that readers are indeed influenced by such framing devices — and we find no evidence that they consider other factors, such as the source, journalist, or the content itself. In addition, we examine the impact of specific framing devices on perceptions of credibility. |  |
| **Sentence-composed summary** | **s10**: **This dataset** was annotated by Mechanical Turk workers who gave ratings for the factuality of **the scoped claims** in each Twitter message.  **s11**: This enables us to build a predictive model of the factuality annotations, with the goal of determining the full set of relevant factors, including **the predicate**, **the source**, **the journalist**, and the content of **the claim** itself.  **s96**: We are unaware of prior work comparing the contribution of linguistic and extra-linguistic predictors (e.g., source and journalist features) for factuality ratings.  **s97**: **This prior work** also does not measure the impact of individual **cues** and cue classes on assessment of factuality.  **s103**: However, this prior work has not explored the linguistic basis of factuality judgments, which we show to depend on **framing devices** such as cue words. | **A**: 2 |
| **B**: 3 |
| Number of dangling anaphors: 9 |
| **Group-composed summary** | **g5**: This paper investigates how linguistic resources and extra-linguistic factors affect perceptions of the certainty of quoted information in Twitter. We present a new dataset of Twitter messages that use FactBank predicates (e.g., claim, say, insist) to scope the claims of named entity sources. This dataset was annotated by Mechanical Turk workers who gave ratings for the factuality of the scoped claims in each Twitter message. This enables us to build a predictive model of the factuality annotations, with the goal of determining the full set of relevant factors, including the predicate, the source, **the journalist**, and the content of the claim itself. However, we find that these extra-linguistic factors do not predict readers’ factuality judgments, suggesting that **the journalist’s own framing** plays a decisive role in the credibility of the information being conveyed. | **A**: 5 |
| **B**: 5 |
| **Number of dangling anaphors:** 2 |
| **Paragraph-composed summary** | **p3**: This paper investigates how linguistic resources and extra-linguistic factors affect perceptions of the certainty of quoted information in Twitter. We present a new dataset of Twitter messages that use FactBank predicates (e.g., claim, say, insist) to scope the claims of named entity sources. This dataset was annotated by Mechanical Turk workers who gave ratings for the factuality of the scoped claims in each Twitter message. This enables us to build a predictive model of the factuality annotations, with the goal of determining the full set of relevant factors, including the predicate, the source, **the journalist**, and the content of the claim itself. However, we find that these extra-linguistic factors do not predict readers’ factuality judgments, suggesting that **the journalist’s own framing** plays a decisive role in the credibility of the information being conveyed. | **A**: 5 |
| **B**: 5 |
| Number of dangling anaphors: 2 |

Table E-26 Standard summary, produced summaries and manual readability annotations of P14-2069

|  |  |  |
| --- | --- | --- |
| **P14-2069** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Emotion lexicons play a crucial role in sentiment analysis and opinion mining. In this paper, we propose a novel Emotion-aware LDA (EaLDA) model to build a domain-specific lexicon for predefined emotions that include anger, disgust, fear, joy, sadness, surprise. The model uses a minimal set of domain-independent seed words as prior knowledge to discover a domain-specific lexicon, learning a fine-grained emotion lexicon much richer and adaptive to a specific domain. By comprehensive experiments, we show that our model can generate a high-quality fine-grained domain-specific emotion lexicon. |  |
| **Sentence-composed summary** | **s10**: However, since a specific word can carry various emotions in different domains, a general-purpose emotion lexicon is less accurate and less informative than a domain-specific lexicon (Baccianella et al., 2010).  **s11**: For **emotion topics**, **the EaLDA model** draws the word distribution from a biased Dirichlet prior Dir(β (e) k ).  **s96**: For each word in **the document**, we decide whether its topic is an emotion topic or a nonemotion topic by flipping a coin with head-tail probability (p (e) , p(n) ), where (p (e) , p(n) ) ∼ Dir(α).  **s97**: for each emotion topic k ∈ {1, . . . , M}, draw ϕ (e) k ∼ Dir(β (e) k ). | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 3 |
| **Group-composed summary** | **g5**: For **emotion topics**, **the EaLDA mode**l draws the word distribution from a biased Dirichlet prior Dir(β (e) k ).  **g5**: for each emotion topic k ∈ {1, . . . , M}, draw ϕ (e) k ∼ Dir(β (e) k ).  **g5**: for each non-emotion topic k ∈ {1, . . . , K}, draw ϕ (n) k ∼ Dir(β (n) ).  **g5**: draw **θ (e)** ∼ **Dir(α (e) )**.  **g5**: draw **θ (n)** ∼ **Dir(α (n) )**.  **g5**: draw w ∼ **Mult(ϕ (e) z (e) )** , **emit** word w. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 8 |
| **Paragraph-composed summary** | **p3**: for each **emotion topi**c k ∈ {1, . . . , M}, draw **ϕ (e) k** ∼ **Dir(β (e) k )**.  **p3**: for each non-emotion topic k ∈ {1, . . . , K}, draw **ϕ (n) k** ∼ **Dir(β (n) )**.  **p3**: draw **θ (e)** ∼ **Dir(α (e) )**.  **p3**: draw **θ (n)** ∼ **Dir(α (n) )**.  **p3**: draw **z (e)** ∼ **Mult(θ (e) )**.  **p3**: draw w ∼ **Mult(ϕ (e) z (e) )** , **emit** word w. | **A**: 0 |
| **B**: 0 |
| Number of dangling anaphors: 13 |

Table E-27 Standard summary, produced summaries and manual readability annotations of P14-2078

|  |  |  |
| --- | --- | --- |
| **P14-2078** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | The disambiguation algorithm presented in this paper is implemented in SemLinker, an entity linking system. First, named entities are linked to candidate Wikipedia pages by a generic annotation engine. Then, the algorithm re-ranks candidate links according to mutual relations between all the named entities found in the document. The evaluation is based on experiments conducted on the test corpus of the TAC-KBP 2012 entity linking task. |  |
| **Sentence-composed summary** | **s57**: If l candidate **URIs** are provided for each **NE**, then **AD** has (l + 4) columns cu,u∈{1,l+4}.  **s82**: **The system** makes use of all the **semantic content** stored in AD to locally improve the precision of each URI annotation in **the document**. | **A**: 0  **B**: 0 |
|  |
| Number of dangling anaphors: 6 |
| **Group-composed summary** | **g34**: In **MDP**, for each Wikipedia **URI** candidate annotation, all the internal links and categories contained in the source Wikipedia document related to this URI are collected. This information will be used to calculate a weight for each of the l candidate URI annotations of each **mention**. For a given **NE**, this weight is expected to measure the mutual relations of a candidate annotation with all the other candidate annotations of NEs in the document. | **A**: 3 |
| **B**: 4 |
| Number of dangling anaphors: 4 |
| **Paragraph-composed summary** | **p17**: In **MDP**, for each Wikipedia **URI** candidate annotation, all the internal links and categories contained in the source Wikipedia document related to this URI are collected. This information will be used to calculate a weight for each of the l candidate URI annotations of each **mention**. For a given **NE**, this weight is expected to measure the mutual relations of a candidate annotation with all the other candidate annotations of NEs in the document. | **A**: 3 |
| **B**: 4 |
| Number of dangling anaphors: 4 |

Table E-28 Standard summary, produced summaries and manual readability annotations of P14-2101

|  |  |  |
| --- | --- | --- |
| **P14-2101** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Latent topics derived by topic models such as Latent Dirichlet Allocation (LDA) are the result of hidden thematic structures which provide further insights into the data. The automatic labelling of such topics derived from social media poses however new challenges since topics may characterise novel events happening in the real world. Existing automatic topic labelling approaches which depend on external knowledge sources become less applicable here since relevant articles/concepts of the extracted topics may not exist in external sources. In this paper we propose to address the problem of automatic labelling of latent topics learned from Twitter as a summarisation problem. We introduce a framework which apply summarisation algorithms to generate topic labels. These algorithms are independent of external sources and only rely on the identification of dominant terms in documents related to the latent topic. We compare the efficiency of existing state of the art summarisation algorithms. Our results suggest that summarisation algorithms generate better topic labels which capture event-related context compared to the top-n terms returned by LDA. |  |
| **Sentence-composed summary** | **s6**: Such top words are usually ranked using the marginal probabilities P(wi |tj ) associated with each word wi for a given topic tj.  **s9**: where the top 10 words ranked by P(wi |tj ) for **this topic** are listed..  **s49**: Given the set of documents C relevant to topic k, we proposed to generate a label of a desired length x from **the summarisation of C**.  **s90**: [htbp] **GS** for Topic Labels {algorithmic} [1] \REQUIRE LDA topics for **TW**, and the LDA topics for **NW** for **category c**.  **s91**: \ENSURE Gold standard topic label for each of the LDA topics for TW.  **s99**: We compared the results of **the summarisation techniques** with the top terms (TT) of a topic as our baseline.  **s100**: These TT set corresponds to the top x terms ranked based on the probability of the word given the topic (p(w|k)) from the topic model. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 7 |
| **Group-composed summary** | **g3**: Our research task of automatic labelling a topic consists on selecting a set of words that best describes the semantics of the terms involved in this topic. The most generic approach to automatic labelling has been to use as **primitive labels** the topn words in a topic distribution learned by **a topic model** such as LDA (Griffiths and Steyvers, 2004; Blei et al., 2003). Such top words are usually ranked using the marginal probabilities P(wi |tj ) associated with each word wi for a given topic tj . This task can be illustrated by considering the following topic derived from social media related to Education.  **g50**: 1) We ran LDA on **TW** and **NW** separately for each category with the number of topics set to 100; 2) We then aligned the Twitter topics and Newswire topics by the similarity measurement of word distributions of these topics (Ercan and Cicekli, 2008; Haghighi and Vanderwende, 2009; Wang et al., 2009; Delort and Alfonseca, 2012); 3) Finally to generate the **GS** label for each aligned topic pair (ti − tj ), we extracted the headlines of the news articles relevant to tj and selected the top x most frequent words (after stop word removal and stemming). | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 5 |
| **Paragraph-composed summary** | **p1**: Our research task of automatic labelling a topic consists on selecting a set of words that best describes the semantics of the terms involved in this topic. The most generic approach to automatic labelling has been to use as **primitive labels** the topn words in a topic distribution learned by **a topic model** such as LDA (Griffiths and Steyvers, 2004; Blei et al., 2003). Such top words are usually ranked using the marginal probabilities P(wi |tj ) associated with each word wi for a given topic tj . This task can be illustrated by considering the following topic derived from social media related to Education.  **p26**: We compared the results of **the summarisation techniques** with the top terms (TT) of a topic as our baseline. These TT set corresponds to the top x terms ranked based on the probability of the word given the topic (p(w|k)) from the topic model | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 3 |

Table E-29 Standard summary, produced summaries and manual readability annotations of P14-2104

|  |  |  |
| --- | --- | --- |
| **P14-2104** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | In this paper we introduce a semantic role labeler for Korean, an agglutinative language with rich morphology. First, we create a novel training source by semantically annotating a Korean corpus containing fine-grained morphological and syntactic information. We then develop a supervised SRL model by leveraging morphological features of Korean that tend to correspond with semantic roles. Our model also employs a variety of latent morpheme representations induced from a larger body of unannotated Korean text. These elements lead to state-of-the-art performance of 81.07% labeled F1, representing the best SRL performance reported to date for an agglutinative language. |  |
| **Sentence-composed summary** | **s15**: We used **this corpus** to develop, train, and test our Korean **SRL** model.  **s21**: Our corpus is roughly similar in size to the PKPB, and taken together, the two Korean corpora now total about half the size of the Penn English PropBank.  **s34**: **Table 1** shows the semantic roles considered in our annotated corpus.  **s39**: We have annotated semantic roles by following the PropBank annotation guideline (Bonial et al., 2010) and by using frame files of the Penn Korean PropBank built by Palmer et al. (2006).  **s110**: We ran Kokoma Korean morpheme analyzer4 on each sentence of **the Donga corpus** to divide words into morphemes to build latent morpheme representations. | **A**: 1 |
| **B**: 1 |
| Number of dangling anaphors: 4 |
| **Group-composed summary** | **g7**: In this paper, we describe a Korean **SRL** system which achieves 81% labeled semantic F1-score. As far as we know, this is the highest accuracy obtained for Korean, as well as any agglutinative language.  **g64**: We first tested on general features in previous work (2nd column in **Table 3**). We achieved 64.83% and 66.88% on the PKPB and our corpus. When the both corpora were combined, we had 64.86%.  **g72**: We augmented our model with all kinds of features (the last column in Table 3). We achieved our best F1-score of 81.07% over **all scenarios** on our corpus. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p41**: We augmented our model with all kinds of features (the last column in **Table 3**). We achieved our best F1-score of 81.07% over **all scenarios** on our corpus.  **p42**: For Korean **SRL**, we semantically annotated a corpus containing detailed morphological annotation. We then developed a supervised model which leverages Korean-specific features and a variety of latent morpheme representations to help deal with a sparsity problem. Our best model achieved 81.07% in F1-score. In the future, we will continue to build our corpus and look for the way to use unsupervised learning for SRL to apply to another language which does not have available corpus. | **A**: 4 |
| **B**: 4 |
| **Number of dangling anaphors:** 3 |

Table E-30 Standard summary, produced summaries and manual readability annotations of P14-2105

|  |  |  |
| --- | --- | --- |
| **P14-2105** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | We develop a semantic parsing framework based on semantic similarity for open domain question answering (QA). We focus on single-relation questions and decompose each question into an entity mention and a relation pattern. Using convolutional neural network models, we measure the similarity of entity mentions with entities in the knowledge base (KB) and the similarity of relation patterns and relations in the KB. We score relational triples in the KB using these measures and select the top scoring relational triple to answer the question. When evaluated on an open-domain QA task, our method achieves higher precision across different recall points compared to the previous approach, and can improve F1 by 7 points. |  |
| **Sentence-composed summary** | **s10**: we train two semantic similarity models: one links a mention from the question to an entity in the KB and the other maps a relation pattern to a relation.  **s11**: The answer to the question can thus be derived by finding the relation–entity triple r(e1, e2) in the KB and returning the entity not mentioned in the question.  **s47**: The mapping between the pattern and the relation in the KB, as well as the mapping between the mention and the entity are determined by corresponding semantic similarity models.  **s83**: We train two CNN semantic models from sets of pattern–relation and mention–entity pairs, respectively. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 0 |
| **Group-composed summary** | **g4**: Leveraging the question paraphrase data mined from the WikiAnswers corpus by Fader et al. (2013), we train two semantic similarity models: one links a mention from the question to an entity in the KB and the other maps a relation pattern to a relation. The answer to the question can thus be derived by finding the relation–entity triple r(e1, e2) in the KB and returning the entity not mentioned in the question. By using a general semantic similarity model to match patterns and relations, as well as mentions and entities, our system outperforms the existing rule learning system, PARALEX (Fader et al., 2013), with higher precision at all **the recall points** when answering the questions in the same test set. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 1 |
| **Paragraph-composed summary** | **p2**: In this paper, we propose a semantic parsing framework tailored to single-relation questions. At the core of our approach is a novel semantic similarity model using convolutional neural networks. Leveraging the question paraphrase data mined from the WikiAnswers corpus by Fader et al. (2013), we train two semantic similarity models: one links a mention from the question to an entity in the KB and the other maps a relation pattern to a relation. The answer to the question can thus be derived by finding the relation–entity triple r(e1, e2) in the KB and returning the entity not mentioned in the question. | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 0 |

Table E-31 Standard summary, produced summaries and manual readability annotations of P14-2106

|  |  |  |
| --- | --- | --- |
| **P14-2106** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | This paper presents experiments with WordNet semantic classes to improve dependency parsing. We study the effect of semantic classes in three dependency parsers, using two types of constituencyto-dependency conversions of the English Penn Treebank. Overall, we can say that the improvements are small and not significant using automatic POS tags, contrary to previously published results using gold POS tags (Agirre et al., 2011). In addition, we explore parser combinations, showing that the semantically enhanced parsers yield a small significant gain only on the more semantically oriented LTH treebank conversion. |  |
| **Sentence-composed summary** | **s0**: This work presents a set of experiments to investigate the use of lexical semantic information in dependency parsing of English.  **s4**: We will apply different types of semantic information to three dependency parsers.  **s6**: Does semantic information in WordNet help dependency parsing found improvements in dependency parsing using MaltParser on gold POS tags.  **s16**: Different parsers can use semantic information in diverse ways.  **s18**: We will run parser combination experiments with and without semantic information, to determine whether it is useful in the combined parsers.  **s98**: Extended parsers, adding semantic information to **the baselines**. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 1 |
| **Group-composed summary** | **g0**: This work presents a set of experiments to investigate the use of lexical semantic information in dependency parsing of English. Whether semantics improve parsing is one interesting research topic both on parsing and lexical semantics. Broadly speaking, we can classify the methods to incorporate semantic information into parsers in two: systems using static lexical semantic repositories, such as WordNet or similar ontologies (Agirre et al., 2008; Agirre et al., 2011; Fujita et al., 2010), and systems using dynamic semantic clusters automatically acquired from corpora (Koo et al., 2008; Suzuki et al., 2009).  **g53**: This work has tried to shed light on the contribution of semantic information to dependency parsing. | **A**: 5 |
| **B**: 5 |
| Number of dangling anaphors: 0 |
| **Paragraph-composed summary** | **p0**: This work presents a set of experiments to investigate the use of lexical semantic information in dependency parsing of English. Whether semantics improve parsing is one interesting research topic both on parsing and lexical semantics. Broadly speaking, we can classify the methods to incorporate semantic information into parsers in two: systems using static lexical semantic repositories, such as WordNet or similar ontologies (Agirre et al., 2008; Agirre et al., 2011; Fujita et al., 2010), and systems using dynamic semantic clusters automatically acquired from corpora (Koo et al., 2008; Suzuki et al., 2009).  **p3**: Is the type of semantic information related to the type of parser? | **A**: 4 |
| **B**: 4 |
| Number of dangling anaphors: 0 |

Table E-32 Standard summary, produced summaries and manual readability annotations of P14-2111

|  |  |  |
| --- | --- | --- |
| **P14-2111** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Tweets often contain a large proportion of abbreviations, alternative spellings, novel words and other non-canonical language. These features are problematic for standard language analysis tools and it can be desirable to convert them to canonical form. We propose a novel text normalization model based on learning edit operations from labeled data while incorporating features induced from unlabeled data via character-level neural text embeddings. The text embeddings are generated using an Simple Recurrent Network. We find that enriching the feature set with text embeddings substantially lowers word error rates on an English tweet normalization dataset. Our model improves on stateof-the-art with little training data and without any lexical resources. |  |
| **Sentence-composed summary** | **s12**: When run on new strings, the activations of the units in the hidden layer at each position in the string are recorded and used as features for training the string transduction model.  **s34**: Once trained **the model** is used to label new strings and **the predicted edit script** is applied to the input string producing the normalized output string.  **s41**: We use **SRNs** to induce **character-level text representations** from unlabeled Twitter data to use as features in the string transduction model.  **s52**: We use **them** to bring in information from unlabeled data into our string transduction model and then train a character-level SRN language model on unlabeled tweets. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g4**: Our model learns sequences of edit operations from labeled data using a Conditional Random Field (Lafferty et al., 2001). Unlabeled data is incorporated following recent work on using character-level text embeddings for text segmentation (Chrupała, 2013), and word and sentence boundary detection (Evang et al., 2013).  **g22**: We use **them** to bring in information from unlabeled data into our string transduction model and then train a character-level **SRN** language model on unlabeled tweets. We run the trained model on new tweets and record the activation of **the hidden layer** at each position as the model predicts the next character. These activation vectors form our text embeddings: they are discretized and used as input features to the supervised sequence labeler as described in Section 3.4. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p1**: Our model learns sequences of **edit operations** from labeled data using a Conditional Random Field (Lafferty et al., 2001). Unlabeled data is incorporated following recent work on using **character-level text embeddings** for text segmentation (Chrupała, 2013), and word and sentence boundary detection (Evang et al., 2013). We train a recurrent neural network language model (Mikolov et al., 2010; Mikolov, 2012b) on a large collection of tweets. When run on new strings, the activations of the units in the hidden layer at each position in the string are recorded and used as features for training the string transduction model.  **p11**: We use a sequence labeling model to learn to label input strings with **edit scripts**. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 3 |

Table E-33 Standard summary, produced summaries and manual readability annotations of P14-2121

|  |  |  |
| --- | --- | --- |
| **P14-2121** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | This paper presents the first computationally-derived scalar measurement of metaphoricity. Each input sentence is given a value between 0 and 1 which represents how metaphoric that sentence is. This measure achieves a correlation of 0.450 (Pearson’s R, p<0.01) with an experimental measure of metaphoricity involving human participants. While far from perfect, this scalar measure of metaphoricity allows different thresholds for metaphoricity so that metaphor identification can be fitted for specific tasks and datasets. When reduced to a binary classification evaluation using the VU Amsterdam Metaphor Corpus, the system achieves an F-Measure of 0.608, slightly lower than the comparable binary classification system’s 0.638 and competitive with existing approaches. |  |
| **Sentence-composed summary** | **s17**: This paper introduces a system for producing a scalar measurement of metaphoricity which places sentences anywhere between 0 (literal) and 1 (highly metaphoric.  **s43**: For the first task, the binary identification task, the metaphoricity of a sentence was computed by taking the percentage of participants who identified it as metaphoric.  **s55**: The second experiment asks participants to label metaphoricity, this time including a distinction between slightly metaphoric and highly metaphoric sentences.  **s57**: The metaphoricity values for this experiment were calculated in the same way the percentage of participants who rated a sentence as highly metaphoric.  **s66**: The metaphoricity value was computed by taking the percentage of participants who identified a sentence as the most metaphoric of **the three given sentences** | **A**: 4 |
| **B**: 5 |
| Number of dangling anaphors: 1 |
| **Group-composed summary** | **g10**: For the first task, the binary identification task, the metaphoricity of a sentence was computed by taking the percentage of participants who identified it as metaphoric.  **g34**: For the purposes of **this evaluation**, the thresh old for metaphor was set independently for each **genre** and tied to the number of sentences containing metaphorically used words, as rated by the annotators of the corpus. Thus, for the number x of metaphors in the genre, the x sentences with the top metaphoricity values were identified as metaphoric. This illustrates the flexibility of **such a scalar approach** to metaphor identification. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p8**: For the first task, the binary identification task, the metaphoricity of a sentence was computed by taking the percentage of participants who identified it as metaphoric. Thus, if all participants agreed that a sentence was metaphoric, then it receives a 1, while if half of the participants agreed then it receives a 0.5. The idea here is that high metaphoricity is consciously available to participants, so that the more agreement there is about metaphor the more the participants are aware of the sentence’s metaphoricity and thus the higher its metaphoricity value should be. The results of this first experiment are summarized in **Table 1** with the mean, standard deviation, and range of the metaphoricity measurements. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 1 |

Table E-34 Standard summary, produced summaries and manual readability annotations of P14-2133

|  |  |  |
| --- | --- | --- |
| **P14-2133** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Do continuous word embeddings encode any useful information for constituency parsing? We isolate three ways in which word embeddings might augment a stateof-the-art statistical parser: by connecting out-of-vocabulary words to known ones, by encouraging common behavior among related in-vocabulary words, and by directly providing features for the lexicon. We test each of these hypotheses with a targeted change to a state-of-the-art baseline. Despite small gains on extremely small supervised training sets, we find that extra information from embeddings appears to make little or no difference to a parser with adequate training data. Our results support an overall hypothesis that word embeddings import syntactic information that is ultimately redundant with distinctions learned from treebanks in other ways. |  |
| **Sentence-composed summary** | **s0**: This paper investigates a variety of ways in which word embeddings might augment a constituency parser with a discrete state space.  **s6**: It has been less clear how (and indeed whether) word embeddings in and of themselves are useful for constituency parsing.  **s55**: For **OOV** words which are not in the dictionary of embeddings, we back off to **the unknown word model** for **the underlying parser**  **s102**: With the goal of exploring how much useful syntactic information is provided by unsupervised word embeddings, we have presented three variations on a state-of-the-art parsing model, with extensions to the out-of-vocabulary model, **lexicon**, and **feature set**. | **A**: 3 |
| **B**: 3 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g1**: It has been less clear how (and indeed whether) word embeddings in and of themselves are useful for constituency parsing.  **g34**: To evaluate **the vocabulary expansion hypothesis**, we introduce a simple but targeted out-of-vocabulary (OOV) model in which every unknown word is simply replaced by its nearest neighbor in the training set. For OOV words which are not in the dictionary of embeddings, we back off to the unknown word model for **the underlying parser.**  **g43**: We use the Maryland implementation of the Berkeley parser as our baseline for **the kernel-smoothed lexicon,** and the Maryland featured parser as our baseline for the embedding-featured lexicon. 1 1 Both downloaded from https://code.google. com/p/umd-featured-parser/ | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 3 |
| **Paragraph-composed summary** | **p13**: To evaluate **the vocabulary expansion hypothesis**, we introduce a simple but targeted out-of-vocabulary (OOV) model in which every unknown word is simply replaced by its nearest neighbor in the training set. For OOV words which are not in the dictionary of embeddings, we back off to the unknown word model for **the underlying parser.**  **p26**: We take the best-performing combination of all of **these models** (based on **development experiments**, a combination of **the lexical pooling model** with β = 0.3, and OOV, both using C&W word embeddings), and evaluate this on the WSJ test set (**Table 2**). We observe very small (but statistically significant) gains with 300 and 3000 train sentences, but a decrease in performance on the full corpus. | **A**: 2 |
| **B**: 2 |
| Number of dangling anaphors: 6 |

Table E-35 Standard summary, produced summaries and manual readability annotations of P14-2135

|  |  |  |
| --- | --- | --- |
| **P14-2135** | **Texts and dangling anaphors** | **Readability Ratings** |
| **Abstract** | Models that learn semantic representations from both linguistic and perceptual input outperform text-only models in many contexts and better reflect human concept acquisition. However, experiments suggest that while the inclusion of perceptual input improves representations of certain concepts, it degrades the representations of others. We propose an unsupervised method to determine whether to include perceptual input for a concept, and show that it significantly improves the ability of multi-modal models to learn and represent word meanings. The method relies solely on image data, and can be applied to a variety of other NLP tasks. |  |
| **Sentence-composed summary** | **s53**: We apply image dispersion-based filtering as follows if **both concepts** in an **evaluation pair** have an image dispersion below a given threshold, both **the linguistic and the visual representations** are included.  **s63**: The filtering approach described thus far improves multi-modal representations because image dispersion provides a means to distinguish **concrete concepts** from more **abstract concepts**.  **s65**: To evaluate the effectiveness of image dispersion as a proxy for concreteness we evaluated our algorithm on a binary classification task based on the set of 100 concrete and 100 abstract concepts A∪C introduced in Section 2. | **A**: 3 |
| **B**: 2 |
| Number of dangling anaphors: 5 |
| **Group-composed summary** | **g25**: We apply image dispersion-based filtering as follows: if **both concepts** in an evaluation pair have an image dispersion below a given threshold, both **the linguistic and the visual representations** are included. If not, in accordance with the Dual Coding Theory of human concept processing (Paivio, 1990), only the linguistic representation is used.  **g43**: We presented a novel method, image dispersionbased filtering, that improves **multi-modal representations** by approximating **conceptual concreteness** from images and filtering model input. The  results clearly show that including more perceptual input in multi-modal models is not always better | **A**: 5 |
| **B**: 4 |
| Number of dangling anaphors: 4 |
| **Paragraph-composed summary** | **p18**: We apply image dispersion-based filtering as follows: if **both concepts** in an evaluation pair have an image dispersion below a given threshold, both **the linguistic and the visual representations** are included. If not, in accordance with the Dual Coding Theory of human concept processing (Paivio, 1990), only the linguistic representation is used. For **both datasets**, we set the threshold as **the median image dispersion**, although performance could in principle be improved by adjusting this parameter. We compare dispersion filtered representations with linguistic, perceptual and standard multi-modal representations (concatenated linguistic and perceptual representations). | **A**: 3 |
| **B**: 2 |
| Number of dangling anaphors: 5 |