Identifying and Resolving Annotation Changes for Natural Language Understanding

Jose Garrido Ramas
Amazon Alexa AI, Germany
jrramas@amazon.de

Giorgio Pessot
Amazon Alexa AI, Germany
pessot@amazon.de

Abdalghani Abujabal
Amazon Alexa AI, Germany
abujabaa@amazon.de

Martin Rajman
EPFL, Lausanne, Switzerland
martin.rajman@epfl.ch

Abstract

Annotation conflict resolution is crucial towards building machine learning models with acceptable performance. Past work on annotation conflict resolution had assumed that data is collected at once, with a fixed set of annotators and fixed annotation guidelines. Moreover, previous work dealt with atomic labeling tasks. In this paper, we address annotation conflict resolution for Natural Language Understanding (NLU), a structured prediction task, in a real-world setting of commercial voice-controlled personal assistants, where (1) regular data collections are needed to support new and existing functionalities, (2) annotation guidelines evolve over time, and (3) the pool of annotators changes across data collections. We devise an approach combining information-theoretic measures and a supervised neural model to resolve conflicts in data annotation. We evaluate our approach both intrinsically and extrinsically on a real-world dataset with 3.5M utterances of a commercial dialog system in German. Our approach leads to dramatic improvements over a majority baseline especially in contentious cases. On the NLU task, our approach achieves 2.75% error reduction over a no-resolution baseline.

1 Introduction

Supervised learning is ubiquitous as a form of learning in NLP (Abujabal et al., 2019; Finkel et al., 2005; Rajpurkar et al., 2016), but supervised models require access to high-quality and manually annotated data so that they perform reasonably. It is often assumed that (1) such annotated data is collected once and then used to train and test various models, (2) the pool of annotators is fixed, and (3) annotation guidelines are fixed (Benikova et al., 2014; Manning, 2011; Poesio and Artstein, 2005; Versley, 2006). In real-world NLP applications e.g., voice-controlled assistants such as Google Home or Amazon Alexa, such assumptions are unrealistic. The assistant is continuously evolving and extended with new functionalities, and hence, changes to annotation guidelines are frequent. The assistant also needs to adapt to language variations over time, both lexical and semantic. Therefore, annotated data needs to be collected regularly i.e., new collections of data at different time points, where the same utterance text can be re-annotated over time. Additionally, the set of annotators might change across collections. In this work, we tackle the problem of resolving annotation conflicts in a real-world scenario of a commercial personal assistant.

To minimize annotation conflicts, the same data point is often labeled by multiple annotators and the annotation with unanimous agreement, or the one with majority votes is deemed correct (Benikova et al., 2014; Bobicev and Sokolova, 2017; Brants, 2000). While such measures ensure the quality of annotations within the same batch, they cannot ensure it across batches at different time points, particularly when the same data point is present in different batches with inevitable changes to annotation guidelines. For detecting and resolving conflicts, two main methodologies have been explored; Bayesian modeling and training a supervised classification model (Hovy et al., 2013; Plank et al., 2014; Snow et al., 2008; Versley and Steen, 2016; Volokh and Neumann, 2011). Both methodologies make certain assumptions about the setting, for example, annotation guidelines and the pool of annotators are fixed, which is not the case for our use case. Additionally, while Bayesian modeling is reasonably efficient for small datasets, it is prohibitively expensive for large-scale datasets with millions of utterances. We adopt a combination of information-theoretic measures and a classification neural model to detect and resolve conflicts.

NLU is a key component in language-based applications, and is defined as the combination of: (1) An Intent Classifier (IC), which classifies an utterance into one of N intent labels (e.g. `PlayMusic`), and (2) A slot labeling (SL) model, which classifies

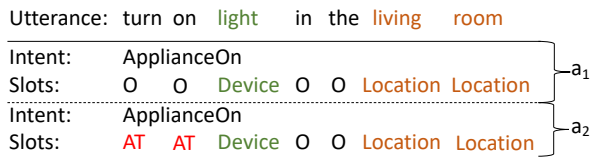

Figure 1: An example utterance with two conflicting annotations, a_1 and a_2. The phrase *turn on* has two conflicting slot labels. AT stands for `ActionTrigger`. Non-entities are labeled with O (i.e., Other).

tokens into slot types, out of a predefined set (e.g. `SongName`) (Goo et al., 2018; Jolly et al., 2020). An example utterance is shown in Figure 1, with two conflicting annotations. In this paper, we consider the task of NLU for personal assistants and assume that utterances arrive at different points in time, and that the annotation guideline evolves over time. The same utterance text, e.g., the one shown in Figure 1, often occurs multiple times across collections, which gives the opportunity to conflicting annotations. Moreover, changes to the annotation guidelines over time lead to more conflicts.

Given an NLU dataset with utterances having multiple, possibly conflicting annotations (IC and SL), our goal is to find the right annotation for each such utterance. To this end, we first detect guideline changes using a maximum information gain cut (Section 3.3). Then we compute the normalized entropy of the remaining annotations after dropping the ones before a guideline change. In case this entropy is low, we simply use majority voting, otherwise, we rely on a classifier neural-based model to resolve the conflict (Section 3.4). Our approach is depicted in Figure 2.

We evaluate our approach both intrinsically and extrinsically, and show improved performance over baselines including random resolution or no resolution in six domains, as detailed in Section 4.

2 Related Work

Annotation conflicts could emerge due to different reasons, be it imprecision in the annotation guideline (Manning, 2011; van Deemter and Kibble, 2000), vagueness in the meaning of the underlying text (Poesio and Artstein, 2005; Recasens et al., 2011, 2010; Versley, 2006), or annotators being careless or inexperienced (Manning, 2011; Hovy et al., 2013). Manning et al. (2011) report, on the WSJ Part-of-Speech (POS) corpus, that 28.0% of POS tagging errors stem from imprecise annotation guideline that caused inconsistent annotations, while 15.5% of the errors are due to wrong gold standard, which could be attributed to careless or inexperienced annotators. In our case, conflicts could occur due to changes to the annotation guidelines and having different, possibly inexperienced, annotators within and across data collections.

Past work on conflict resolution has assumed that data is collected once and then used for model training and testing. Consequently, the proposed methods to detect and resolve conflicts are geared towards this setting (Benikova et al., 2014; Manning, 2011; Poesio and Artstein, 2005; Recasens et al., 2011, 2010; van Deemter and Kibble, 2000; Versley, 2006). In our scenario, we deal with an ever-growing data which is collected across different data collections at different time points. This increases the likelihood of conflicts especially with frequent changes to the annotation guideline. In Dickinson and Meurers (2003), an approach is proposed to automatically detect annotation errors in gold standard annotations for POS tagging using n-gram tag variation i.e., looking at n-grams occurring in the corpus with multiple tagging.

Bayesian modeling is often used to model how reliable each annotator is and to correct/resolve wrong annotations (Hovy et al., 2013; Snow et al., 2008). In Hovy et al. (2013), they propose MACE, an item-response based model, to identify *spammer* annotators and to predict the correct underlying labels. Applying such models is prohibitively expensive in our case due to the large amount of utterances we deal with. Additionally, our annotator pool changes over time. A different line of work has explored resolving conflicts in a supervised classification setting, similar to our approach for resolving high normalized entropy conflicts. Volokh and Neumann (2011) use an ensemble of two off-the-shelf parsers that re-annotate the training set to detect and resolve conflicts in dependency treebanks. Versley et al. (2016) use a similar approach on out-of-domain treebanks. Finally, Plank et al. (2014) introduce the inter-annotator agreement loss to ensure consistent annotations for POS tagging.

Intent classification and slot labeling are two fundamental tasks in spoken language understanding, dating back to early 90's (Price, 1990). With the rise of task-oriented personal assistants, the two tasks got more attention and progress has been made by applying various deep learning techniques (Abujabal and Gaspers, 2019; Goo et al., 2018;

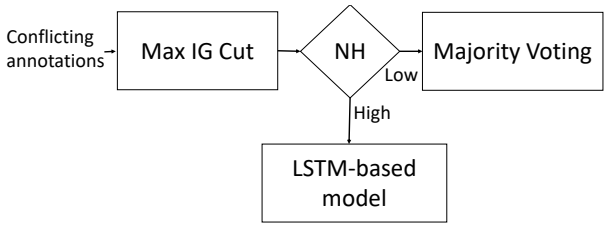

Figure 2: Our approach for conflict resolution. Given conflicting annotations, we first use the Max Information Gain (*IG*) Cut to detect changes in annotation guidelines. Then, low entropy conflicts are resolved using majority voting. High entropy conflicts are resolved using a classifier LSTM-based model.

Jolly et al., 2020; Mesnil et al., 2013; Zhang and Wang, 2016). While we focus on resolving annotation conflicts for NLU with linear labeling i.e., intent and slot labels, our approach can be still used for other more complex tree-based labeling e.g., labeling dependency parses or ontology trees (Chen and Manning, 2014), with the minor change of replacing the task-specific neural LSTM-based classification model. We plan to investigate this in the future.

3 Annotation Conflict Resolution

3.1 Overview

Given multiple conflicting annotations of an utterance, our goal is to find the right annotation. We assume that annotations arrive at different points in time and that the same utterance can be re-annotated over time. Moreover, we assume that annotators might differ both within and across data collections, that each annotation is time stamped, and that there is always one correct annotation. Our pipeline for conflict resolution is depicted in Figure 2. Given an utterance with conflicting annotations, we first detect guideline changes using a maximum information gain cut. Then we compute the normalized entropy of the remaining annotations i.e., without the annotations before a guideline change. In case this entropy is low, we simply use majority voting, otherwise, we rely on a classifier model to resolve the conflict.

A natural choice to easily resolving annotation conflicts is to use majority voting. However, we argue that this is not sufficient for our use case, where (1) regular data collection and annotation are required at different time points, and (2) changes to annotation guideline are frequent. We use the normalized entropy to *detect* whether there is high or low disagreement among annotations. In the extreme case where the normalized entropy is 1, majority voting gives a random output and any model that performs better than random will be better than majority voting in resolving conflicts. In our experiments we show that, for high normalized entropy values, the classifier model significantly outperforms majority voting.

Note that our conflict resolution pipeline does not drop utterances with wrong annotations, but rather replaces the wrong annotations with the correct ones. We do so to avoid changing the data distribution.

We apply our pipeline to training data only. The test set is of higher quality compared to the train set as each collection of test set data is annotated multiple times and we use the most recent test set collection.

3.2 Normalized Entropy

Entropy measures the uncertainty of a probability distribution (Yang and Qiu, 2014). Given an utterance present N times in the dataset and annotated in K distinct ways, each occurring n_i times such that $\sum_{i=1}^{K} n_i = N$, we define the *normalized empirical entropy* of the list of conflicting annotations A, $NH(A)$ as:

$$NH(A) = \frac{-\sum_{i=1}^{K} \frac{n_i}{N} * \log\left(\frac{n_i}{N}\right)}{\log K}, \; for \; K > 1$$

For example, assume an utterance u with three distinct annotations; a_1, a_2 and a_3. Then, the list A corresponds to $\{a_1, a_2, a_3\}$, $K = 3$, and p_i of each annotation corresponds to its relative frequency in the dataset ($\frac{n_i}{N}$) (Mahendra et al., 2014).

In this work, we harness normalized entropy (NH) to determine whether majority voting should be used. NH is a value between 0 and 1, where the higher it is, the harder the conflict. In the edge case of a uniform distribution, where NH is 1, majority voting gives a random output. Therefore, in such cases, we do not rely on majority voting for conflict resolution but rather on a classification model. We use the normalized entropy over entropy as the latter increases as K increases when the distribution is uniform. For example, assume $K = 3$ and distribution is uniform, then entropy is $H = \log 3$, and $NH = 1$. If $K = 2$ and distribution is uniform, then $H = \log 2$ and $NH = 1$, and so on. When the distribution is uniform (and thus majority voting will be outperformed by a model regardless

of K), NH takes its maximum value of 1, while H increases as K increases (Kvålseth, 2017).

3.3 Changes in Annotation Guideline: Max Information Gain Cut

We rely on max information gain cut to find out if there was a change in the annotation scheme that caused a conflict, and to identify the exact date d of the change. Let us assume the relatively common case that there is exactly one relevant change in the guideline. Then, we aim to split the annotations of an utterance to two lists; one list containing annotations prior to the change, and the other one containing annotations after the change.

Inspired by methods used for splitting on a feature in decision trees (Mahendra et al., 2014), we harness *information gain* (IG) to determine the date to split at. Concretely, given a list B of chronologically ordered annotations for the same utterance, and their corresponding annotation dates, we choose the date d that maximizes IG. If the value of IG is larger than a threshold IG_0, we deem the annotations prior to d incorrect. The higher the IG is, the more probable the annotations prior to d to be incorrect. We define a boolean variable D which is true if the date of an annotation comes after d, and false otherwise. It divides the list of annotations B to two sublists, B_b of size N_b of annotations before date d, and B_a of size N_a of annotations after date d. We compute IG as follows:

$$IG(B, D) = NH(B) - NH(B|D), where$$

$$NH(B|D) = \frac{N_b * NH(B_b) + N_a * NH(B_a)}{N}$$

We use the normalized entropy (NH) for IG computation, as shown in the equation above. As a result, IG is no longer strictly positive.

In the case of changes in the annotation guideline, there will be high disagreement among annotations before and after the change, and thus, $NH(B)$ will be high. Moreover, annotations before the change will agree among each other, and similarly, for annotations after the change. Therefore, $NH(B|D)$ will be low. Then $IG(B, D)$ takes its maximum value at the date of the guideline change, and annotations after this date, which belong to the latest guideline, are correct. For example, for the following date-ordered annotations; $\{a_1$(03-2019), a_1(07-2019), a_1(08-2019), a_2(10-2019), a_2(11-2019), a_3(12-2019), a_2(01-2020), a_2(02-2020)$\}$, spliting at d = (08-2019) yields the highest IG value, as shown in Figure 3. This indicates that there was a change in the annotation of this utterance on 08-2019. Hence, a_1 annotation is deemed wrong. In Section 4.2, we empirically prove that for high IG values, a large percentage of annotations occurring in the first half of the Max IG Cut split is incorrect, whereas a large percentage of annotations in the second half is correct.

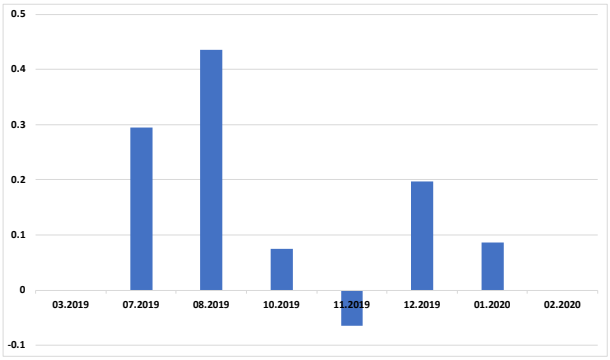

Figure 3: IG values at each date. The split at d =08-2019 has the highest IG value. We cannot split at the first and last dates.

After the split, NH is computed for the remaining annotations i.e., annotations after d. If NH is less than a threshold NH_0, we assign the utterance the annotation with maximum frequency (i.e., *majority voting*). In the example above, NH is low after the split, and the conflict is resolved by changing all annotations (i.e., a_1 and a_3) to a_2. Our reasoning is that, when NH is high, majority voting will likely be outperformed by an alternative model (LSTM-based method, explained next) as there is high disagreement between the annotators. Note that we do not drop any utterances, we replace wrong annotations with the correct ones.

3.4 High Entropy Conflicts: LSTM

To make classification in the ambiguous high NH cases, we use a supervised classifier trained on the unambiguous examples from our data, in this case an LSTM-based neural model (Hochreiter and Schmidhuber, 1997). For the following list of annotations, $\{a_1, a_2, a_3, a_2, a_1, a_3\}$, no split with IG greater than a threshold can be found, and $NH = 1$. For such utterances, we rely on a neural model to estimate the probability of each annotation i.e., a_1, a_2, and a_3. Then we assign the annotation with highest probability to the utterance. Concretely, we use the model of Chiu et al. (2016), a bidirectional word-level LSTM model with a character-based

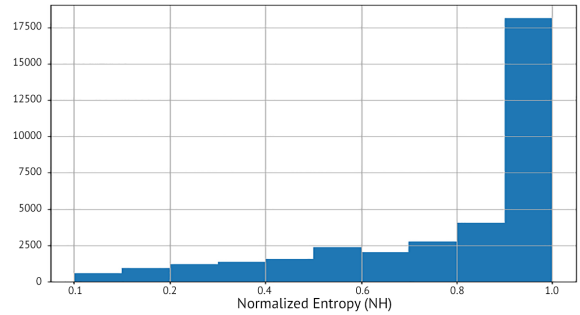

Figure 4: Histogram of conflicts in the training data. Most conflicts have high entropy.

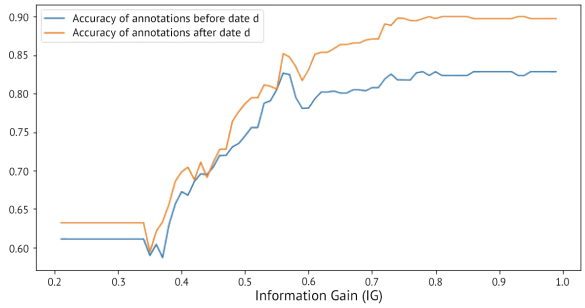

Figure 5: Accuracy of the rule change detection method described in Section 3.3. For high IG values, the accuracy of annotations after a date d, at which there is a guideline change, is 90%, while the accuracy of annotations before d is over 80%.

CNN layer. A softmax layer is used on top of the output of the bidirectional LSTM, which computes a probability distribution over the output slot labels for a given input token. We extend the model to a multi-task setting to support IC by concatenating the last hidden states of the Bi-LSTM, and passing them to a softmax layer, similar to Yang et al. (2016). We harness the probabilities of the output of the softmax layer and compute the final probability of the annotation by multiplying the probability of each of its slots and of the intent.

4 Experiments

In this section we evaluate our method both intrinsically and extrinsically.

4.1 Setup

Data. We use a real-world dataset of a commercial dialog system in German, belonging to six different domains covering different, macro-purposes like, for instance, musical or movies requests. For the purpose of IC and SL, domains are treated as separate datasets. Utterances were manually transcribed and annotated with domain, intent and slot labels across many different batches at different points of time. In total we have 3.5M and 560K training and testing utterances, respectively. The percentage of conflicts in the training data varies across domains, ranging from 4.9% to 10.9%. Most conflicts are of high entropy, as shown in Figure 4. The test set is of higher quality compared to the train set as each collection of test set data is annotated twice. Generally, the test set has lower number of conflicts compared to the train set. We do not resolve the conflicts in the test data to avoid artificial inflation of results.

LSTM model. For high entropy conflicts, we use a single layer network for the forward and the backward LSTMs whose dimensions are set to 256. We use Glove pretrained German word embeddings (Pennington et al., 2014) with 300 dimensions. For the CNN layer, character embeddings were initialized randomly with 25 dimensions. We used a mini-batch Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. We tried different optimizers with different learning rates (e.g., stochastic gradient descent), however, they performed worse than Adam. We also applied Dropout of 0.5 to each LSTM output (Hinton et al., 2012). For training, we use the data described above (i.e., 3.5M utterances) after applying the Max IG Cut and majority voting to resolve low entropy conflicts, as described in Section 3.3. High-entropy conflicts are left unresolved. After 10 epochs, training is terminated. After training is done, the model is used for conflict resolution for high entropy cases.

4.2 Intrinsic Evaluation

To asses the quality of our method, an expert linguist is asked to resolve 490 conflicts in two different domains e.g., Music. The linguist is asked to use the latest annotation guideline. On average, we have 12.6 utterances per conflict, with a total number of 6173 utterances for the 490 conflicts. The maximum number of utterances of a conflict is 181. On the annotation side, the maximum number of unique annotations of a conflict is 8, while the average number is 2.35 (Table 1).

We used our pipeline to resolve the 490 conflicts that were resolved by the linguist, where 229 conflicts out of the 490 were resolved with the LSTM model, which means that 46.7% of the conflicts were of high normalized entropy ($\geq NH_0 = 0.75$).

	#Utterances per Conflict	#Unique Annotations per Conflict
Min	2	2
Average	12.6	2.35
Max	181	8
Total	6173	1151

Table 1: Statistics on the 490 conflicts used for our evaluation.

Guideline change detected	120
Resolved with LSTM model	229
Resolved with majority voting	261

Table 2: Out of the 490 conflicts, 229 were resolved with the LSTM model, while 261 conflicts were resolved with majority voting.

The remaining 261 conflicts were resolved with majority voting. 120 out of the 490 conflicts had at least one guideline change (Table 2).

Max IG cut. For those conflicts with guideline changes we evaluate, after splitting the list of annotations at date d, whether the annotations after d are correct (a^i_{after}), and whether the annotations before d are incorrect (a^i_{before}). To this end, for each conflict with $IG \geq 0.2$, we compare each annotation after and before d with the ground-truth annotation (a_{gt}) provided by the linguist. a^i_{after} annotations should be correct, therefore, accuracy is 1 if a^i_{after} agrees with a_{gt}, and 0 otherwise. On the other hand, a^i_{before} annotations should be incorrect, and hence, accuracy is 1 if a^i_{before} does not agree with a_{gt}, and 0 otherwise. We compute the average accuracy over a^i_{after} annotations and the average accuracy over a^i_{before} annotations for each conflict. We also compute the average across those conflicts with the same IG value.

We depicted the results in Figure 5. For high IG values, high accuracies are achieved for annotations after and before a split at a date d. For example, at IG = 0.9, the accuracy of annotations before d is almost 0.83, while the accuracy of annotations after d is 0.90. This shows that our max IG cut method was able to identify the right date d to split the list of annotations at for the majority of conflicts with guideline changes. We set IG_0 to 0.4.

Majority Voting vs. LSTM. We evaluate the resolution of the 490 conflicts with the LSTM-based model and majority voting at different levels of NH. For each conflict, we apply the max IG cut and then resolve it using both methods of majority voting and LSTM. We then compare the final annotation each method delivers as correct with that delivered by the linguist. If both agree, then accuracy is 1, and 0 otherwise. For each NH value, we compute the average accuracy of the set of 50 conflicts with closest NH.

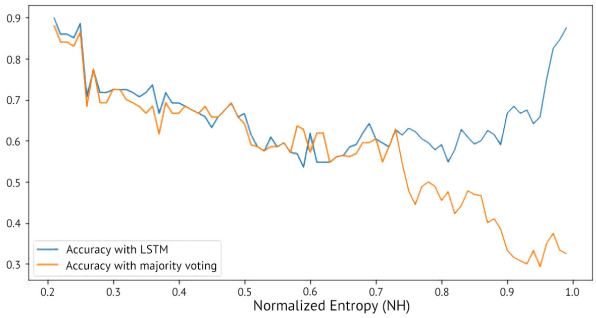

Figure 6: Accuracy with majority voting (orange) and with the LSTM-based method (blue) on the 490 conflicts with respect to ground-truth resolution provided by the linguist. For high values of NH, the LSTM-based model performs better than majority voting.

As expected, the accuracy with majority voting significantly drops with high entropy conflicts, as shown in Figure 6. The LSTM-based model becomes more accurate as NH increases, reaching the highest accuracy in the case where $NH = 1$. In the training data, 29.3% of conflicts have $NH = 1$. As seen in the figure, accuracy diverges at $NH = 0.75$, which we use as NH_0. That is, if $NH \geq 0.75$, we use the LSTM-based model, and majority voting otherwise. For NH below 0.75, both majority voting and the LSTM-based model behave similarly, however, we use majority voting for low entropies as it is more intuitive.

4.3 Effect on NLU

To evaluate our method extrinsically on the downstream task of NLU, we trained a multi-task LSTM-based neural model for intent classification and slot labeling on the 3.5M utterances after resolving annotation conflicts using our proposed method (Figure 2). Architecture-wise, the model is similar to the one we use for conflict resolution, described in Section 3.4. We compared this model with two baseline models trained as follows:

1. **NoResolution**: this model was trained on the full training data without conflict resolution (i.e., 3.5M utterances).

Method	Error Rate (Rel. Change)
Random Resolution	0.55%
Our Pipeline	**2.75%**

Table 3: Results on the NLU task. Our pipeline achieved 2.75% relative change in error rate with respect to the NoResolution baseline.

2. **Rand**: We trained this model with conflicts resolved by choosing one annotation randomly.

The three models were tested on the same test set described above (560K utterances). We report the relative change in error rate with respect to the NoResolution model. The error rate is defined as the fraction of utterances in which there is at least an error either in IC or in SL.

Results are shown in Table 3. Overall, random conflict resolution slightly reduced the error rate with 0.55% relative change on average across domains, while our method achieved 2.75% error reduction. For each of the six domains, resolving conflicts with our method improves performance over random resolution and over no resolution. In one domain, a reduction in error rate of 4.7% is observed. For five domains, the difference in performance passes a two-sided paired t-test for statistical significance at 95% confidence level.

5 Conclusion

In this paper, we tackled the problem of annotation conflicts for the task of NLU for voice-controlled personal assistants. We presented a novel approach that combines information-theoretic measures and an LSTM-based neural model. We evaluated our method on a real-world large-scale dataset, both intrinsically and extrinsically.

Although we focused on the task of NLU, our conflict resolution pipeline could be applied to any manual annotation task. In the future, we plan on investigating how the choice of the task-specific classification model affects performance. Moreover, we plan to study annotation conflict resolution for other NLP tasks e.g., PoS tagging and dependency parsing.

Acknowledgements

We thank Melanie Bradford and our anonymous reviewers for their thoughtful comments and useful discussions.

References

Abdalghani Abujabal and Judith Gaspers. 2019. Neural Named Entity Recognition from Subword Units. In *Proc. Interspeech 2019*, pages 2663–2667.

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 307–317. Association for Computational Linguistics.

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. NoSta-d named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland. European Language Resources Association (ELRA).

Victoria Bobicev and Marina Sokolova. 2017. Inter-annotator agreement in sentiment analysis: Machine learning perspective. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 97–102, Varna, Bulgaria. INCOMA Ltd.

Thorsten Brants. 2000. Inter-annotator agreement for a German newspaper corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 740–750. ACL.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4:357–370.

Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 363–370. The Association for Computer Linguistics.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 753–757. Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. 2013. Learning whom to trust with MACE. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1120–1130. The Association for Computational Linguistics.

Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 10–20, Online. International Committee on Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tarald O. Kvålseth. 2017. On normalized mutual information: Measure derivations and properties. *Entropy*, 19(11).

M.S. Mahendra, E.J. Neuhold, A.M. Tjoa, and I. You. 2014. *Information and Communication Technology: Second IFIP TC 5/8 International Conference, ICT-EurAsia 2014, Bali, Indonesia, April 14-17, 2014, Proceedings*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing - 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part I*, volume 6608 of *Lecture Notes in Computer Science*, pages 171–189. Springer.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, pages 3771–3775. ISCA.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 742–751. The Association for Computer Linguistics.

Massimo Poesio and Ron Artstein. 2005. The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky@ACL 2005, Ann Arbor, MI, USA, June 29, 2005*, pages 76–83. Association for Computational Linguistics.

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990*. Morgan Kaufmann.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Marta Recasens, Eduard Hovy, and M. Antònia Martí. 2010. A typology of near-identity relations for coreference (NIDENT). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Marta Recasens, Eduard Hovy, and M Antònia Martí. 2011. Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua*, 121(6):1138–1152.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October*

2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 254–263. ACL.

Kees van Deemter and Rodger Kibble. 2000. On coreferring: Coreference in MUC and related annotation schemes. *Comput. Linguistics*, 26(4):629–637.

Yannick Versley. 2006. Disagreement dissected: Vagueness as a source of ambiguity in nominal (co-) reference. In *Ambiguity in Anaphora Workshop Proceedings*, pages 83–89.

Yannick Versley and Julius Steen. 2016. Detecting annotation scheme variation in out-of-domain treebanks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Alexander Volokh and Günter Neumann. 2011. Automatic detection and correction of errors in dependency treebanks. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 346–350. The Association for Computer Linguistics.

Jiping Yang and Wanhua Qiu. 2014. Normalized expected utility-entropy measure of risk. *Entropy*, 16:3590–3604.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2993–2999. IJCAI/AAAI Press.