2008 IEEE Workshop on Spoken Language Technology (SLT 08)

C^5

D. Suendermann, J. Liscombe, K. Evanini, K. Dayanidhi, R. Pieraccini

SpeechCycle, Inc., New York City, USA
david@speechcycle.com

COMPENDIUM

The annotation of hundreds of thousands of utterances for the training of statistical utterance classifiers requires a careful quality assurance procedure to make the data consistent and reliable. In this paper, we present five methods to analyze different aspects of annotated data to ensure their Completeness, Consistency, Correlation, Congruence and to avoid Confusion—collectively referred to as C^5.

Index Terms— annotation, statistical utterance classification, quality assurance

1. COMMENCEMENT

Statistical utterance classification was proposed a decade ago to introduce natural language in automated dialog applications [1]. In order to build a statistical classifier, one requires a number of utterances typical of the task and a class associated to each of the utterances conveying their semantics. These utterance-class mappings are used to train a statistical model that later serves as a knowledge base to a classifier that is to map a new, and unlabeled, utterance to one of the set of possible classes. As demonstrated e.g. in [2], the more training data available, the better classification rates can be achieved. State-of-the-art classifiers are trained on hundreds of thousands of utterances [3].

The annotation of such a number of utterances may keep several annotators busy for several months. While it is well-known that the combination of different peoples' annotations suffers from inter-labeler disagreement [4, 5], intra-labeler disagreement can also be significant. This is due not only to the fact that annotators tend to map certain utterances to different classes depending on the time of the day, their mood, and whether they had a coffee before, but there are also objective reasons for such inconsistencies:

- Statistical utterance classifiers are an integral component of a dialog system which uses utterance classification to take certain actions. Thus, the semantic meaning of an utterance depends on the dialog context in which it occurs. Paradigm changes in the dialog logic may lead to a change of the canonical assignment between utterances and classes.
- More drastically, such a paradigm change can lead to the creation of new classes or the collapsing or elimination of existing classes.
- Most difficult utterances are those which show a potential overlap among several existing classes or which seem to belong to none of the given classes.
- Utterances may be very vague or exhibit expressions that do not fit into the expected vocabulary, thus forcing annotators to heavily interpret vague language.

Practically speaking, all these sources of inconsistency require a revision of the entire corpus at regular intervals. On the one hand, such revision rounds can barely be applied to the entirety of utterances due to limits of time and resources. On the other hand, a partial re-annotation—limited to certain classes or utterances containing certain key words—will hardly cover all inconsistencies in the data. As aforementioned, we also face a considerable inter-labeler disagreement that comes with the fact that hundreds of thousands of utterances cannot be labeled by a single person in a reasonable time frame. These effects may lead to a disastrous annotation situation which finally would suggest to limit the data used for training to a couple of thousand (but clean) utterances rather than the indomitable beast whose unreliability is more harmful than its size is helpful.

This paper investigates five techniques that help to escape this dilemma. They deal with the Consistency of data, its Completeness, the Correlation among annotators, Confusions in the data and, finally, the Congruence between annotations and predicted classes—for their leading letter's coincidence baptized C^5.

2. COMPLETENESS

The paradigm "there is no data like more data" is one of the driving forces of statistical speech and language processing. E.g., the performance of speech recognizers or automatic language translation does not seem to get to a saturation point even with very large amounts of training data [6, 7]. It is natural that this paradigm was deemed valid also for the training of statistical utterance classifiers.

As mentioned in Section 1, the application of the utterance classifier referred to in this paper aims at providing the correct interpretation of a caller's natural language utterance in the framework of a dialog system. The caller's speech utterance is first processed by a large-vocabulary continuous speech recognizer whose utterance hypothesis is then classified by a statistical classifier (details below). In order to train the classifier, speech utterances are collected in a production

Patent pending.

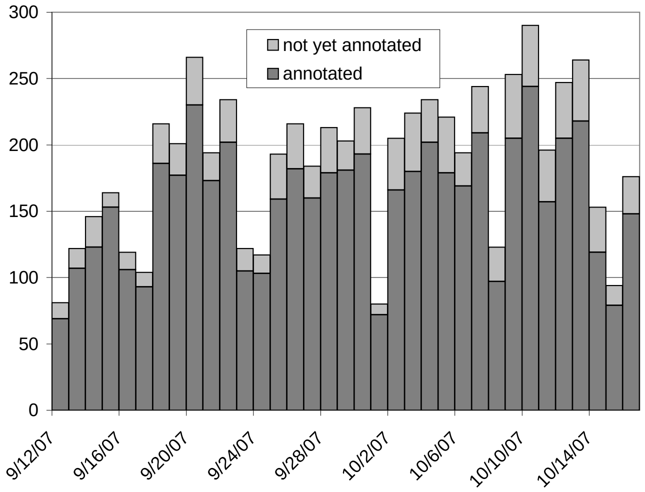

Fig. 1. Effect of non-complete annotation. The example shows the daily transcription/annotation volume of a certain application collected over 35 days. Out of 6521 utterances, 5530 were annotated, i.e. 84.4%.

system where a preliminary utterance classifier (mostly a rule-based grammar) or a simple data collection module is implemented. The applications discussed in this publication process millions of calls per month. Only a fraction of these calls is transcribed and an even lower fraction is annotated due to limitations of human resources for these processes being mostly manual.

In most cases, the number of transcribed utterances is growing faster than that of annotations. Due to the above formulated paradigm, the annotators would try to process as many utterances as possible. Therefore, the desired goal would seem to be to concentrate primarily on frequent and easy utterances whose transcriptions are made available on a daily basis. This means, however, that a certain percentage of utterances remains unprocessed every day, as can be seen in Figure 1.

After a reasonable number of utterances has been annotated, this data would be split into train and test data, the former used to build the classifier, the latter for assessing its performance in batch experiments. Such experiments usually produce satisfying results suggesting a high accuracy of the classifier. We noticed, however, that adopting this approach led to problems. As soon as such a classifier went into production and started taking live calls, its performance decreased.

Why was that? The reason was that we omitted a part of the data (in the example of Figure 1 around 15%) for training and test, namely that data which did not appear to produce quick results, i.e., less frequent utterances and difficult cases. Since the test data also omitted these cases, this negligence was not obvious in the batch experiments. In production, however, around 15% of all utterances for the example application belonged to the omitted type of utterances which have not been used for training and, consequently, resulted in misrecognition in the majority of these cases.

scenario	training utterances	accuracy
incomplete	97,237	62.6%
complete	25,756	68.9%

Table 1. Comparing the performance of incompletely vs. completely annotated data for training a statistical utterance classifier.

To see what impact the absence (or, vice versa, the completeness) of data has on the performance of a statistical utterance classifier, we conducted a number of experiments out of which we select only one prototypical case due to the limited focus of this paper. In this example, we trained a statistical classifier based on the Naïve Bayes approach with boosting [8] on utterances collected from a dialog application for cable TV troubleshooting [9].

One of these classifiers was built according to the greedy paradigm "there is no data like more data"—the annotators were provided 151,184 transcribed utterances and were asked to annotate as many utterances as possible within a four week time frame. After the time was over, they had completed 97,237 annotations (64%).

The other classifier was trained based on the thorough paradigm "there is no data like complete data"—a similar amount of transcriptions was provided, but the annotators were asked to work on a day-by-day basis, i.e., they were supposed to complete the utterances collected over a one-day time period before starting with the next. This resulted, after four weeks, in only 25,756 annotations (17% of the transcriptions).

Then, an experiment was carried out in which approximately 8000 (completely) annotated utterances distinct from the training data were used as a test set. Large vocabulary speech recognition was carried out on these utterances, and the resulting word string was processed by the above described statistical classifier. The classification accuracy was measured as the number of correctly classified acoustic events divided by the total number of acoustic events, i.e., also nonsense utterances, background noise, and the like were taken into consideration.

The results are displayed in Table 1. Although the training data of the complete paradigm comprises only about a quarter of the other scenario's data, it outperforms the latter by 6.3% absolute which equals a relative error reduction of 16.8%. This example clearly shows that the 'tail' of the utterances and classes that is, the less frequent candidates can indeed have a significant impact on the classifier's performance.

3. CORRELATION

Although we have seen that the complete-data rule may be more powerful than the more-data rule, it occurred to us that, finally, the joint hypothesis "there is no data like more complete data" holds true. Hence, in order to further push the performance of a certain application, we relied on a team of up to five annotators working at full time to annotate more

κ	A1	A2	A3	A4	A5	average
A1		0.85	0.59	0.82	0.75	0.75
A2	0.85		0.56	0.80	0.77	0.75
A3	0.59	0.56		0.58	0.51	**0.56**
A4	0.82	0.80	0.58		0.71	0.73
A5	0.75	0.77	0.51	0.71		0.69

Table 2. An example of examining inter-labeler correlation using the kappa statistic.

than 300,000 utterances according to 250 distinct classes (for details see [3]). Different annotators, however, have different opinions about how to label things—sometimes, it is deemed impossible to find a final agreement on the exact class where certain utterances belong due to differences in annotation styles. To isolate subsets of the annotators whose approaches achieve a high level of agreement, we asked them to label the same set of utterances independently of each other and then applied the kappa statistic to determine the level of inter-labeler correlation [10]. $\kappa > 0.7$ is usually considered a sufficient correlation for many tasks.

Table 2 shows values for the kappa statistic for an example set of 1000 utterances of the same domain we used in Section 2 for each of the possible labeler combinations. Here, the intra-annotator comparison featuring the trivial value of $\kappa = 1$ is omitted. Additionally, the table contains the average of the values of each row which, in this case, bears a clear pattern: Annotators A1, A2, A4, and A5 show a rather high agreement, whereas A3's annotation style seems to follow a different direction. This discrepancy may be resolved by intensively training low-performing annotators or discarding them from the present project.

4. CONSISTENCY

Having selected a group of annotators adhering the desired annotation style, we still do not produce a correlation of $\kappa = 1$ among them. According to the authors' experience, the annotators do not even agree with themselves on certain (complicated, vague, or ambiguous) utterances.

To quantify this effect and help annotators to find cases of inter- and intra-labeler inconsistency, we set up a procedure which investigates whether *identical* utterances are associated with more than one class. These cases are corrected (if mistakes due to oversight), or they are subject to a discussion involving annotators, dialog application designers, and speech scientists.

In a second step, another consistency comparison is carried out taking *similar* utterances into account by matching the *bags of words* of the analyzed utterances. This representation is used to reduce redundant information by performing the following steps:

- Stop words are removed according to a list including 38 function words.
- The remaining words are stemmed using the Porter stemmer algorithm [11].
- Multiple occurrences of words are eliminated.
- The order of words is regularized by alphabetic sort.

utterance	class	count
need to be turned on	BoxWontTurnOn	2
i need it turned on	ServiceNoService	5
it needs to be turned on	BoxWontTurnOn	3
needs to be turned on	BoxOther	2

Table 3. Example of an inconsistent annotation of utterances determined by bag-of-word matching.

Table 3 shows an example case of similar utterances mapped to several classes detected by means of bag-of-word matching. Again, this example is taken from the utterance classifier at the open-ended prompt of a cable TV troubleshooting application.

Consistency analysis serves several purposes:

- It helps to remove annotation errors and normalize the data in order to assure quality and achieve highest classification performance of the utterance classifier.
- The rate of confusions per newly annotated utterance is a measure of the task complexity and of the familiarity of an annotator with the current task.
- It helps detect cases of major confusion which indicate that classes should be redefined, collapsed, or split or that annotation instructions are ambiguous.
- Overall, consistency analysis serves as a training tool for annotators and is used to get several annotators on the same page and detect cases of high uncertainty.

5. CONFUSION

We mentioned that consistency analysis helps to detect confusions in annotated data. However, there are types of confusions inherent to the classifier design which may not be visible to the annotators. Similar wordings might convey clearly different meanings to a human being, but to a machine-based probabilistic classifier working on automatic speech recognition output, such utterances might very well be ambiguous. A simple example are the utterances "yes... no!" and "no... yes!" which in a yes/no context are annotated as *no* and *yes*, respectively, since the caller's attitude is derived from his latest pronouncement. The classifier used in this work, however, does not consider word order, consequently, both utterances are identical to its knowledge.

In order to derive a picture of where the classifier's weak points are, we provide completely annotated and consistency-checked data to train an initial classifier and apply it to a distinct set of randomly selected and likewise annotated test utterances. Looking at the confusion matrix of such an experiment points to areas of major confusion which are then subject to more careful inspection. The example of Table 4 refers

		annotated class			
		OOG	no	operator	yes
predicted class	OOG	106	19	**26!**	13
	no	15	240	2	1
	operator	2	0	22	0
	yes	13	3	**12!**	346

Table 4. Confusion matrix of a classification experiment for a yes/no/operator scenario; numbers are counts of confusions.

to a yes/no scenario where we also want to catch operator requests. We see that a large percentage of operator requests is assumed to be out-of-grammar (OOG) or confused with *yes*.

6. CONGRUENCE

In Section 1, we mentioned that, for the initial building of a statistical classifier, a reasonable number of utterances of the target domain is required. In many conditions, such utterances can be collected by bootstrapping a manual grammar based on sets of expressions expected in the respective context and using this grammar at the first place in a live system. Utterances processed in this initial framework are collected, transcribed, annotated, and a first statistical classifier is built and tested on annotated reference data. As soon as the statistical classifier outperforms the manual version, the latter is replaced by the former. In regular intervals, new classifiers are built, tested, and implemented in the live application when their performance is significantly higher than that of their predecessors.

The rule-based grammar used as initial step is often not only a trivial collection of a few utterances expected in the current context, but may exhibit a complex structure incorporating several manually tuned sub-grammars with a significant vocabulary (such as grammars handling requests for operator, repetition, holding, help), loops of phrases, typical affixes callers use, and so on. The number of different utterances described by such a grammar can be infinite (due to loops or recursion). Since it is not clear to what extent the manual grammar's vocabulary reflects real callers' language in the current context, it is not used to train the statistical classifier. However, it can be used to *test* annotations, since it establishes a ground truth. Whatever parse a rule-based grammar returns for an utterance must be identical to the class the annotator maps it to. Otherwise, either the annotator made a mistake, or the rule-based grammar was erroneous, the latter being extremely rare to the authors' experience. Consequently, the congruence between rule-based grammar parse and annotated class is another means of assuring quality of annotation.

Table 5 shows two example scenarios for which rule-based grammars served as initial classifier. The percentage of utterances for which the rule-based grammar returns a parse is rather high for the yes/no/operator example (introduced in Section 5) featuring only four classes. This coverage decreases, the more unpredictable the caller's language becomes. The second example comes from a context where callers are asked for the type of their modems. The answers tend to be more natural and conversational than in a yes/no context which explains the lower coverage. Nonetheless, more than half of all utterances have a rule-based counterpart to their annotations.

grammar	number of classes	rule-based coverage
yes/no/operator	4	77.8%
modem type	28	57.3%

Table 5. Example of coverage of utterances parsable by the initial rule-based grammar for two scenarios.

7. CONCLUSION

In this paper, we have shown how C^5 analysis can be used to assure quality of annotations for the training of statistical utterance classifiers. It also includes methods for training annotators, providing means for self-control and -learning, for selecting and adjusting annotators in a team, and for assessing the overall complexity of an annotation task with respect to the set of classes involved. Some of the presented methods have a direct impact on the accuracy of the utterance classifier (completeness, consistency) while others help to make annotations more efficient and reliable.

8. CROSS-REFERENCES

[1] A. Gorin, G. Riccardi, and J. Wright, "How May I Help You?" *Speech Communication*, vol. 23, no. 1/2, 1997.

[2] K. Evanini, D. Suendermann, and R. Pieraccini, "Call Classification for Automated Troubleshooting on Large Corpora," in *Proc. of the ASRU*, Kyoto, Japan, 2007.

[3] D. Suendermann, P. Hunter, and R. Pieraccini, "Call Classification with Hundreds of Classes and Hundred Thousands of Training Utterances ... and No Target Domain Data," in *Proc. of the PIT*, Kloster Irsee, Germany, 2008.

[4] T. Lander, B. Oshika, J. Carlson, T. Durham, and T. Bailey, "Analysis of Interlabeler (Dis)Agreement in Phonetic Transcriptions of Multiple Languages," *Journal of the Acoustical Society of America*, vol. 100, no. 4, 1996.

[5] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Ess-Dykema, and M. Meteer, "Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech," *Computational Linguistics*, vol. 26, no. 3, 2000.

[6] G. Evermann, H. Chan, M. Gales, B. Jia, D. Mrva, P. Woodland, and K. Yu, "Training LVCSR Systems on Thousands of Hours of Data," in *Proc. of the ICASSP*, Philadelphia, USA, 2005.

[7] T. Brants, A. Popat, P. Xu, F. Och, and J. Dean, "Large Language Models in Machine Translation," in *Proc. of the EMNLP'07*, Prague, Czech Republic, 2007.

[8] C. Elkan, "Boosting and Naive Bayesian Learning," UCSD, San Diego, USA, Tech. Rep., 1997.

[9] K. Acomb, J. Bloom, K. Dayanidhi, P. Hunter, P. Krogh, E. Levin, and R. Pieraccini, "Technical Support Dialog Systems: Issues, Problems, and Solutions," in *Proc. of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, Rochester, USA, 2007.

[10] A. Rosenberg and E. Binkowski, "Augmenting the Kappa Statistic to Determine Interannotator Reliability for Multiply Labeled Data Points," in *Proc. of the HLT/NAACL*, Boston, USA, 2004.

[11] M. Porter, "An Algorithm for Suffix Stripping," *Program*, vol. 14, no. 3, 1980.