Active Annotation in the LUNA Italian Corpus of Spontaneous Dialogues

Christian Raymond[1], **Kepa Joseba Rodriguez**[2], **Giuseppe Riccardi**[3]

[1] L.I.A.,University of Avignon, France
[2] Piedmont Consortium for Information Systems (CSI-Piemonte). Italy
[3] Department of Information Engineering and Computer Science. University of Trento. Italy
christian.raymond@univ-avignon.fr, kepajoseba.rodriguez@csi.it, riccardi@dit.unit.it

Abstract

In this paper we present an active approach to annotate with lexical and semantic labels an Italian corpus of conversational human-human and Wizard-of-Oz dialogues. This procedure consists in the use of a machine learner to assist human annotators in the labeling task. The computer assisted process engages human annotators to check and correct the automatic annotation rather than starting the annotation from un-annotated data. The active learning procedure is combined with an annotation error detection to control the reliablity of the annotation. With the goal of converging as fast as possible to reliable automatic annotations minimizing the human effort, we follow the active learning paradigm, which selects for annotation the most informative training examples required to achieve a better level of performance. We show that this procedure allows to quickly converge on correct annotations and thus minimize the cost of human supervision.

1. Introduction

The aim of the LUNA project is to investigate the problem of spontaneous speech understanding in the context of conversational systems engaged in complex tasks such as the problem-solving paradigm.

Three steps are considered for the Spoken Language Understanding (SLU) process: generation of semantic concept tags, semantic composition into conceptual structures and context sensitive validation. The SLU modules will be trained and evaluated on the LUNA corpus and applied to different conversational systems in Italian, French and Polish.

In this paper, we present the semantic annotation procedure we are following on an Italian corpus. This corpus consists of human-human spontaneous dialogues recorded in the call center of the help desk facility of the Consortium for Information Systems of the Piedmont. The aim of our semantic annotation procedure is to speed up the manual annotation of the corpus and to make more reliable the annotation (Tur et al., 2003; Vlachos, 2006). This procedure consists in using a statistical learner to annotate automatically transcribed files at the semantic level and to generate automatically annotated files in the input format of the annotation tool: human annotators have just to check and correct these annotations instead of starting from scratch. In order to converge as fast as possible to reliable automatic annotations and so minimizing the human effort, this procedure follows the active learning paradigm which selects for annotation the most informative examples and thus reduces the number of supervised training examples needed to achieve a given level of performance. The active learning procedure is coupled with an annotation error detection to assure more reliable annotation.

We present in section 2. the LUNA corpus and in section 3. the specific corpus and the semantic annotations we are talking about in this paper. We introduce briefly the Active Learning paradigm in the section 4.1. and the annotation error detection paradigm in the section 4.2.. The section 5. describes our annotation procedure and presents the first results.

This work was partially funded by the European Commission - LUNA project (contract n°33549) and Marie Curie Excellence Grant ADAMACH project (contract n°022593).

2. The LUNA Spoken Dialogue corpus

The corpus is being collected and annotated with the target of annotating 1000 human-human and 8100 human-machine dialogues in Italian, French and Polish for different application domains.

Here we present an overview of the annotation levels of the LUNA corpus. A more detailed description of the annotation scheme and some examples have been published in (Rodriguez et al., 2007).

2.1. Morphosyntactic annotation

The transcribed material is being annotated with Part of Speech tags, morphosyntactic information and segmented based on syntactic constituency. For the POS-tags and morphosyntactic features, we follow the recommendations made in (EAGLES, 1996).

2.2. Domain-attribute level

At this level semantic segments are being annotated following a similar approach to the used for the French MEDIA dialogue corpus (Bonneau-Maynard and Rosset, 2003). Domain knowledge is organized in a concept dictionary for each application domain. The concept dictionary contains:

- Concepts: corresponding to the attributes of the annotation
- Values
- Constraints on the admissible values.

2.3. Predicate structure level

For the annotation of the predicate structure we use a FRAMENET-like approach (Baker et al., 1998). Based on domain knowledge we define a set of frames for each domain.

2.4. Coreference

The annotation of coreference follows a scheme close to the used in the annotation of dialogues of the TRAINS corpus in ARRAU (Poesio and Artstein, 2008). Markables are annotated with giveness and relatedness to previously mentioned objects.

2.5. Dialogue acts

The annotation of dialogue acts is based on the annotation of predicate structure. We annotate each utterance using a multidimensional annotation scheme partially based in the DAMSL (Allen and Core, 1997).

3. The Italian LUNA Corpus

3.1. General description of the data

The Italian corpus what is being currently transcribed and annotated consists of two different data sets: a set of human-human spontaneous dialogues and a set of Wizard of Oz dialogues.
The general structure of the dialogues is as follows:

1. One of the participants – usually the operator (human or wizard) – opens the dialogue.
2. The operator presents him/herself and asks for the identity of the caller.
3. The operator asks for the problem.
4. The caller explains the problem and both dialogue partners collaborate to find the source of the trouble.
5. The way to solve the problem can be as follows:
   (a) Both dialogue participants collaborate to solve the problem.
   (b) The operator solves the problem alone or tells the caller what is necessary to be done.
6. Both dialogue participants close the dialogue.

3.2. The human-human dialogue data

The human-human dialogue data described in Table 1 consists of spontaneous dialogues recorded in the call center of the help desk facility of the Consortium for Information Systems of the Piedmont (CSI Piemonte[1]).
The recorded dialogues have two dialogue participants, a caller –public worker of the region Piedmont– and an operator of the help desk facility. The main topics of the dialogues are software and hardware problems and related administrative issues. Since these dialogues are spontaneous there are other minor topics, like small talks about other persons, holidays, etc.
As usual in spontaneous dialogues there is a high frequency of interruptions, overlapped contributions, use of cut-off phrases and ungrammatical sentences.

3.3. The Wizard of Oz data

The WoZ dialogue data is being currently recorded in experimental settings in the installations of the CSI-Piemonte. The dialogues of this data set (described in Table 2) are related to different problems with the hardware.

[1] http://www.csi.it

Transcribed dialogues	180
Time (min.)	495.29 ($\overline{x} = 2.75$ *min)*
Number of turns	9074 ($\overline{x} = 50$ *turns)*
Number of words	66290 ($\overline{x} = 368$ *words)*
Number of different words	4715
Number of annot. segments	17462 ($\overline{x} = 97$ *segments*)

Table 1: Description of the human-human data

Transcribed dialogues	249
Time (min.)	130.7 ($\overline{x} = 32$ *sec)*
Number of turns	1525 ($\overline{x} = 6$ *user-turns)*
Number of words	12420 ($\overline{x} = 50$ *words)*
Number of different words	1467
Number of annot. segments	3885 ($\overline{x} = 16$ *segments*)

Table 2: Description of the Wizard of Oz data (only user turns)

3.4. Morphosyntactic annotation

The transcribed data is annotated with Part of Speech and morphosyntactic features on the word level and the words grouped in syntactic chunks using the Chaos Parser ((Basili et al., 1999)).

3.5. Semantic annotation on the attribute-value level

After an analysis of a set of dialogues we defined a hierarchy of 55 concept names and constraints for the possible values. This representation was used to build the concept dictionary used for the annotation.
Some of the main categories of the annotation are:

- Software applications
- Hardware components
- Network components
- Persons: First and last names, professional categories
- Actions that are relevant to identify or solve the problem
- Kinds of documents used
- Identification codes of computers and documents
- Locations: institutions and companies, sections, addresses, web-sites, telephone numbers. etc.
- Temporal expressions

We use this concept dictionary to annotate a first set of 140 dialogues on the domain attribute level as presented in the example (1)[2]. The tool used for the annotation is Semantizer (Bonneau-Maynard and Rosset, 2003) (fig. 1), a tool that was previously used for the annotation of the MEDIA corpus.

(1) **Operator:** sto guardando [lex=filler] l'
[avete aperta]$_{concept1}$ [stamattina]$_{concept2}$
<concept1 action:open>
<concept2 temp-partOfDay:morning>
Caller: sí
Operator: [undici]$_{concept3}$ [trentanove]$_{concept4}$

[2] Translation: **O:***I'm looking [filler] did you open it on the morning?* // **C:** *yes* // **O:** *11 39* // **C:** *if you want to have my RWS-ID 13 835* // **O:** *let us see if* // **C:** *I have open it* // **O:** *you are offline*

<concept3 number-cardinal:11>
<concept4 number-cardinal:39>
Caller: se vuole [la mia RWS]$_{concept5}$
[tredici ottocentotrentacinque]$_{concept6}$ forse
<concept5 code-typ:rws>
<concept6 code-value:13835>
Operator: vediamo se
Caller: te l' [ho aperta]$_{concept7}$ io
<concept7 action:open>
Operator: siete [fuori rete]$_{concept8}$ proprio
<concept8 problem:off_line>

In a fist step we annotated manually a set of 15 dialogues. The semantic segments identified for the manual annotation were produced by concatenation of the chunks produced in the previous level of the annotation. We used this annotation to train a first model in order to be able to perform a semi-automatic annotation of the corpus as presented in the next section.

4. Our Active Annotation principle

We implement an Active Annotation approach in order to reduce the human effort. This approach is based on statistical methods to automatically pre-annotate the data and thus facilitate the human annotator's job. Our approach is based on two paradigms:

1. the **Active Learning** paradigm: it consists to an iterative procedure which selects at each turn the most informative examples to be annotated and thus help our Active Annotation procedure to produce better automatic annotation at each turn;

2. the **annotation error detection**: detect likely erroneous annotation in order to be supervised again by the human annotators. We believe that it could be a double advantage: improve the performance of our statistical methods and help the annotators to avoid some mistake.

4.1. Active Learning

The Active Learning (AL) paradigm consists in the selection of the most informative examples for manual annotation and thus reduces the number of supervised training examples needed to achieve a given level of performance. We use an uncertainty-based AL method (Lewis and Catlett, 1994) which selects for labeling the examples that the learner is least confident about. To use this method, we need a learner and an associated confidence measure. The choice of one or the other is not crucial, however in our situation where we process manual transcription: we do not have real-time constraints nor the need to be robust to the recognition errors. The discriminant algorithms in this situation are accurate (Raymond and Riccardi, 2007) and able to integrate many different knowledge sources. In addition to these abilities, Conditional Random Fields (CRF) (Lafferty et al., 2001) provide the conditional probability over the whole annotation given the observation which can be exploited as confidence measure for the automatic annotation uncertainty (Symons et al., 2006). Since we are annotating dialogue by dialogue from the human annotators side, we need to select full dialogues instead of isolated turns. We extend the turn confidence measure given by the CRF to a dialogue confidence measure which is basically the average of the confidence measures for each turn in the dialogue. We use in this work an open source implementation of CRF (Kudo,).

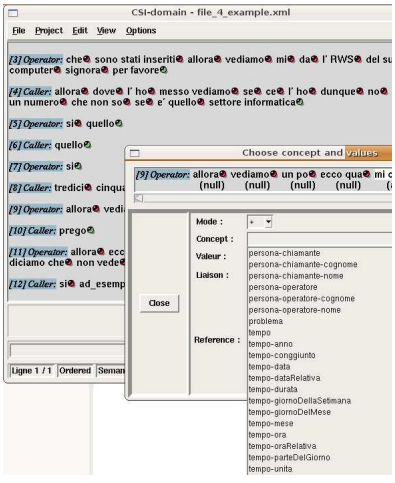

Figure 1: Semantizer Screenshot

4.2. Annotation error detection

Annotation error detection is crucial since annotation error impact significantly the statistical learners performances. The main idea is to detect exceptional elements checking the training set under the control of the statistical algorithm, the examples receiving low confidence are likely to be erroneous or hard examples. In (Abney et al., 1999) they use the highest weighted examples by the boosting algorithm, in (Nakagawa and Matsumoto, 2002) they use the weight assigned by their SVM classifier, in (Raymond and Riccardi, 2008) they use the conditional probability provided by the CRF.

5. Active Annotation

We implement an Active Annotation approach (figure 3) in order to reduce the human effort. This approach is based on statistical methods to automatically pre-annotate the data and thus facilitate the human annotator's job. As detailed in section 3.5. the annotation concerns to the semantic attribute/value representation. For the automatic annotation of the corpus we split the problem in two subtasks:

1. the detection and classification of semantic segments,

2. the extraction of the possible values for the attributes.

The automatic methods used are detailed in the next sections.

5.1. Conditional Random Fields

A conditional random field is defined by a dependency graph G and a set of features f_k to which are associated weights λ_k. The conditional probability of an annotation

given an observation is given by:

$$p(y|x) = \frac{1}{Z(x)} exp(\sum_{c \in C} \sum_{k} \lambda_k f_k(y_c, x, c))$$

$$with$$

$$Z(x) = \sum_{y} exp(\sum_{c \in C} \sum_{k} \lambda_k f_k(y_c, x, c))$$

Semantic features, (*a priori* defined) concept relations, *etc.*, are encoded in the model using these functions. In most case, the features are binary functions returning 1 if there is a match, 0 if not. These features take in parameter the values taken by the random variables (y_c) of the clique (c) to which they apply, and also the *whole* observation x. The weights λ_k associated to each features are the parameters of the model. Learning a CRF is to compute the weights λ_k.

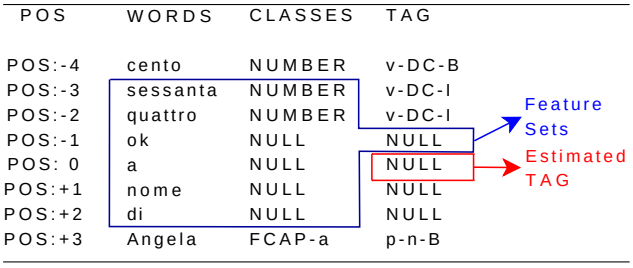

Figure 2: Feature set used for training CRF

5.2. Segmentation and classification problem

The model μ is a CRF tagger used to do concept segmentation and classification. The sequence labeling problem is solved by BIO representation (Ramshaw and Marcus, 1995): each word in the sequence is associated with the corresponding concept together with the B (Begin) or I (Inside) markers to make the concepts boundaries explicit[3]:

tags :vDC-B vDC-I vDC-I null null null null p-n-B
words: cento sessanta quattro okay a nome di Angela

μ is trained using a traditional first order dependency graph. Features are the indicators for specific words and their corresponding generalization class in a window [-3, 2] around the decision state. Generalization classes are MONTH, DAY, DIGIT_NUMBER, ORDINAL, see figure 2. Since we are working on manual transcription, features corresponding to transcriptions protocol are introduced: *e.g.* words with first letter capitalized or all letters capitalized. These features permit to obtain models with better generalization power.

5.3. Value extraction

The CRF tagger produces the concept segmentation and classification. To produce the normalized value for each concept, we use two methods depending of the concept type; examples are available in table 3. We distinguish 2 types of concepts:

1. the first type is concepts with potential infinite/huge number of values (*e.g.* numbers) or potentially not observed in training set (dialogues annotated) while it is easy to produce the exhaustive list (*e.g.* dates): in this situation the value extraction is done by applying a hand-crafted grammar rule set covering the exhaustive list of possible values,

2. the second type contains the remaining concepts: the value extraction is done by a classifier. In this case, no supervision is necessary. The new introduced value by the human annotators will be covered in the next active turn. The classifier chosen is BoosTexter (Schapire and Singer, 2000) an implementation of the boosting algorithm.

method	attribute	chunk	value
Classifier	computer-componentHardware	del mio computer	pc
	azione	cancelli	cancellare
	problema	dei problemi su intranet	problema_rete
Grammar	codice-valoreDiCodice	trentuno duemilasei	312006

Table 3: Example of concepts with attribute, chunk, value and the method chosen to produce the value

5.4. Annotation procedure

Following the procedure detailed in figure 3, we start with N manually annotated dialogues randomly selected (step 1) to build a first model μ, In each AL turn, the dialogues in the unlabeled part S_U are automatically annotated (step 2a). A batch of k dialogues for which the model μ is less confident about is selected (S_k) and provided in the format of the annotation tool used, Semantizer. Then S_k is presented for human control/correction instead of annotating them from scratch. The manually corrected files are then added to the set of training data. A new model μ is trained and the process is repeated.
At the same time, the statistical algorithm re-annotates the annotations used for training and the difference between automatic and manual annotation permit to exhibit at each turns annotation errors/ambiguities, see bold part in the figure 3.
A discriminant learner like one using CRF tends to fit the training data, and the model obtained should be able to reproduce the same annotation as the annotation used for training. In our framework we are comparing the automatic annotation produced with a model trained with the original annotations. If the original manual annotation differs from the automatic annotation, we have 2 cases:

1. **the manual annotation is correct:** an example hard to learn. If the example is hard to learn is because the model does not have the feature needed to discriminate between the erroneous concept and the good one. Many times the feature needed is very intuitive and could be added to the model easily. For example, we find that the model did many confusion between the concepts "application-software" which could be supported by the words "Lotus Notes",

[3] vDC=valoreDiCodiche & p-n=persona-nome

1. Train a model μ using small amount of N manually annotated dialogues from scratch randomly selected (S_L)
2. while (labeler/data available)
   (a) Use μ to automatically annotate the unannotated part of the corpus (S_U) and to produce Semantizer files
   (b) Rank automatically annotated examples (S_U) according to the confidence measure given by μ
   (c) Select a batch of k dialogues with the lowest score (S_k)
   (d) Ask for human control/correction on S_k
   (e) **Use μ to automatically annotate S_L and produce S_L^a**
   (f) **Look at the difference between S_L and S_L^a**
      i. HARD EXAMPLE TO LEARN: **add new features when training μ**
      ii. ANNOTATION AMBIGUITIES: **Hire human annotators to disambiguate S_L**
   (g) $S_L = S_L + S_k$
   (h) Train a new model μ with S_L

Figure 3: Active annotation procedure: the non-bold faced part correspond to the traditional active learning algorithm, the bold part correspond to the annotation error detection strategy

	# dialogues	# turns	# attrib.	# values
Wizard of oZ	249	1417[4]	36	487
Human-Human	180	9074	50	1511

Table 4: Statistics about attribute/value annotated data

"Outlook", *etc.* and "person-name" which could be "Roberto", "Marco", *etc.*. It's clear that, it's because these words are associated with the same class (FIRST_LETTER_CAPITALIZED) and the local context used in our model is not discriminant. Many Italian names are finishing by letters "a,i,o", so a new feature taking into account this information was introduced in our model and made it more accurate.

2. **the manual annotation is erroneous:** correct the manual annotation.

6. Evaluation

6.1. Evaluation from the procedure side

We evaluated the capacities of the automatic procedure to produce correct annotation. After each active annotation turn we compare the automatic annotation and the final annotated data (*i.e.* automatic annotation corrected by human annotators). The comparison is done at two levels: first the ability of the automatic procedure to produce the correct segmentation and classification, secondly the ability to produce the correct normalized value for an attribute:

- to evaluate the accuracy of our system to produce segmentation and classification, we consider as entities the couples *sequence of words/concept attribute*, including the null[5] concept. For example the utterance in section 5.2. is represented as:

  `valoreDiCodiche[cento_sessanta_quattro] null[okay_a_nome_di] persona-nome[Angela].`

  Then we compute the entity error rate in the same way as the word error rate, all entities in a turn are aligned using the Levenshtein distance and the entity error rate is the rapport between the sum of errors (*i.e.* Insertions, Deletions, Subsitutions) and the number of entities in the manual annotation:

$$Entity\ error\ rate = \frac{\#Ins.\#Subs.\#Del.}{\#ref.entities}$$

- to evaluate the value extraction accuracy, we consider as entities the concepts themselves, *i.e.* the couples *attribute/value* excluding the null concept. For example the utterance in section 5.2. is represented as:

  `valoreDiCodiche[164] persona-nome[angela]`

The statistics are available in table 5 for WoZ dialogues and table 6 for Human-Human dialogues. These tables 5 and 6 present the performance of the automatic annotation for each active annotation turn. Let us focus on table 6, in the first turn (first line of the table), we used in step $N = 10$ dialogues annotated manually to train a model, we used it to annotate $k = 10$ transcribed dialogues composed of 561 turns. After correction by human annotators we compare the manual and automatic annotation: the model produced automatic annotations containing 62.2% of erroneous turns in terms of segmentation+classification (it means that 62.2% of these turns needed to be corrected) the entity error rate was $\frac{1531}{2151}$ =71.2%, 1531 of erroneous entities on a total of 2151. In terms of attribute/value, 73.9% of entities have to be corrected. It could be interpreted as high error rate, but the training data used in the first turn is very small and third of the turns has been correctly annotated. The percentage of annotation to be corrected decrease at each turn and in the last step only about third of them has to be corrected except the normalized values where 53.6% have to be corrected; this high number is explained by the high number of value, about 1500.

6.2. Evaluation from the annotators side

A further goal achieved in this experiment is the reduction of the time that human annotators need to annotate a dialogue file. At the beginning of the annotation process the annotators needed in average between 80 and 90 minutes to annotate a dialogue file. After the third annotation loop the annotators needed between 25 and 35 minutes to correct the output of the classifier. In following loops the time needed by the annotators to perform the annotation task remains constant. A possible explanation is the impossibility to supervise the annotation of a dialogue file in less time regardless how good is the quality of the automatic annotation.

7. Conclusion

In this paper we present the LUNA Italian corpus of spontaneous human-human dialogues. We present the semantic annotation at domain entities level we are currently doing. The task is especially difficult because domain entities

[5] the concept associated with the no-meaning words

		Automatic annotation							
			segmentation+classification			attribute/value			
Act. turn	train size in turn	# turns	turns error rate	entity error rate	error vs. all	turns error rate	entity error rate	error vs. all
1	200	200	99.5%	59.2%	1580/2669	99.0%	57.8%	964/1669
2	400	200	77.0%	44.4%	434/978	84.5%	52.2%	302/579
3	600	200	54.0%	39.3%	297/756	59.0%	41.8%	205/490
4	800	400	7.8%	6.4%	60/944	32.0%	28.6%	151/528
5	1200	217	0.0%	0.0%	0/265	2.8%	11.8%	6/51

Table 5: Statistics on active annotation for WoZ dialogues in terms of segmentation&classification and attribute/value extraction at each active annotation turn

		Automatic annotation							
				segmentation+classification			attribute/value		
Act. turn	train size in dialog.	# dia.	# turns	turns error rate	entity error rate	error *vs.* all	turns error rate	entity error rate	error *vs.* all
1	10	10	561	62.2%	71.2%	1531/2151	62.4%	73.9%	873/1181
2	20	10	517	51.5%	59.5%	1090/1831	55.9%	81.0%	790/975
3	30	10	490	44.5%	54.0%	866/1605	50.2%	72.0%	633/879
4	40	20	1378	43.7%	51.0%	2224/4359	47.5%	71.9%	1635/2274
5	60	20	1547	39.4%	45.7%	2164/4732	48.9%	77.7%	1932/2487
6	80	20	1257	35.0%	42.4%	1497/3533	44.6%	77.3%	1369/1771
7	100	40	2915	32.7%	37.5%	3076/8204	40.5%	63.1%	2674/4237
8	140	40	2103	27.7%	34.5%	1865/5398	33.0%	53.6%	1467/2736

Table 6: Statistics on active annotation for Human-Human dialogues in terms of segmentation&classification and attribute/value extraction at each active annotation turn

can be realized in a fragmentary fashion done to disfluencies, truncated words and phrases and other features of the spontaneous language. We propose an active annotation framework following the active learning paradigm which uses statistical algorithm to pre-annotate semantically transcribed files in order to speed-up and make easier the human annotation process. In the actual phase of the experimentation the framework seems to offer good results.

In the near future we plan to experiment with an extension of the actual approach to other levels of annotation like coreference, which involves the recognition and classification of relations between entities.

8. References

Steven Abney, Robert E. Schapire, and Yoram Singer. 1999. Boosting applied to tagging and PP attachment. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 38–45. 4.2.

James Allen and Mark Core. 1997. Draft of DAMSL: Dialog act markup in several layers. Unpublished manuscript. 2.5.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90, Morristown, NJ, USA. Association for Computational Linguistics. 2.3.

Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 1999. Lexicalizing a Shallow Parser. In *Conference sur le Traitment Automatique des Langues Naturelles (TALN99)*. 3.4.

Hélène Bonneau-Maynard and Sophie Rosset. 2003. A semantic representation for spoken dialogues. In *Proceedings of Eurospeech*, Geneva. Semantizer available at: *http://www.limsi.fr/Individu/hbm*. 2.2., 3.5.

EAGLES. 1996. Recomendations for the Morphosyntactic Annotation of Corpora. 2.1.

Taku Kudo. Crf++. Available online at `http://crfpp.sourceforge.net/`. 4.1.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289. 4.1.

David Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of ICML-94, 11th International Conference on Machine Learning*, pages 148–156, New Brunswick, US. 4.1.

Tetsuji Nakagawa and Yuji Matsumoto. 2002. Detecting errors in corpora using support vector machines. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 4.2.

Massimo Poesio and Ron Artstein. 2008. Further developments in anaphoric annotation: the ARRAU corpus. In *Proc. of LREC*, Marrakesh. 2.4.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey. Association for Computational Linguis-

tics. 5.2.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Interspeech*, Antwerp, Belgium, August. 4.1.

Christian Raymond and Giuseppe Riccardi. 2008. Learning with noisy supervision for spoken language understanding. In *ICASSP*, pages 183–188, Las Vegas, USA. (Accepted). 4.2.

Kepa Joseba Rodriguez, Stefanie Dipper, Michael Götze, Massimo Poesio, Giuseppe Riccardi, Christian Raymond, and Joanna Rabiega-Wiśniewska. 2007. Standoff coordination for multi-tool annotation in a dialogue corpus. In *Linguistic Annotation Workshop*, Prague. http://www.aclweb.org/anthology-new/W/W07/W07-1524.pdf. 2.

Robert E. Schapire and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168. Available online at `http://www.cs.princeton.edu/~schapire/boostexter.html`. 2

Christopher T. Symons, Nagiza F. Samatova, Ramya Krishnamurthy, Byung H. Park, Tarik Umar, David Buttler, Terence Critchlow, and David Hysom. 2006. Multicriterion active learning in conditional random fields. In *ICTAI: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 323–331, Washington, DC, USA. IEEE Computer Society. 4.1.

Gokhan Tur, Mazin Rahim, and Dilek Hakkani-Tür. 2003. Active labeling for spoken language understanding. In *Eurospeech*, Geneva. 1.

Andreas Vlachos. 2006. Active Annotation. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction*. 1.