Aspects of Named Entity Processing

Michael Levit, Patrick Haffner, Allen Gorin, Hiyan Alshawi, Elmar Nöth

AT&T Labs-research and University Erlangen-Nuremberg
levit@informatik.uni-erlangen.de

Abstract

In this paper we investigate the utility of three aspects of named entity processing: *detection*, *localization* and *value extraction*. We corroborate this task categorization by providing examples of practical applications for each of these subtasks. We also suggest methods for tackling these subtasks, giving particular attention to working with speech data. We employ Support Vector Machines to solve the detection task and show how localization and value extraction can successfully be dealt with using a combination of grammar-based and statistical methods.

1. Introduction

Named entity processing has been granted much attention by several speech research groups [1, 2, 3]. For several years now identification of named entities (NE) has been the subject of a lot of academic but also practically oriented research. The importance of the problem led to the need for standardization and motivated extensive work on the named entity definitions. Today's standard NE-categories by MUC-7 [4] contain definitions for three basic types of named entities: ENAMEX (proper names, acronyms, etc), TIMEX (temporal expression) and NUMEX (numerical expressions, monetary expressions and percentages). In this paper, we take on the procedural aspect of named entity processing. So far, the most effort in this research field was focused on the issue of named entity identification, a task in which one must detect the boundaries of named entities in text (or, in the case of spoken language processing, in the ASR-output), determine the type (name, location etc.) of the named entities in the delimited areas and possibly return the extracted text as a final product of the identification [4]. One issue that such a conceptual formulation seems to neglect is the differentiation between wording and meaning, crucially important for the language understanding applications. While one can often assume a one-to-one correspondence between the two for proper names, this certainly doesn't hold for named entities like dates. Another largely ignored question concerns the issue of the presence of at least one named entity of the given type in the utterance.

We propose a categorization of NE-tasks into *detection*, *localization* and *value extraction* and explain differences and interconnections among these subtasks. We also suggest novel methods for solving each of them. In particular, we use the SVM-classifier to perform the detection task and, based on its outcome, turn on the localization module implemented as an error-tolerant composition of finite state transducers.

Another important aspect of named entity processing addressed in this paper is working with imperfect data. Unlike text documents with preserved orthography, capitalization and other spelling-specific clues, the speech signal does not maintain any other named entity witnesses but their acoustic representation. Thus, algorithms successfully working with spelling features on text (e.g. [5]) can not be applied here. Besides, speech data is prone to noise and susceptible to recognition errors; this is why some flexibility must be integrated in the identification process. From these facts, some authors drew the conclusion of impracticality of handcrafted grammars for named entity extraction from speech [2], while others tried to combine stochastic NE-grammars with handcrafted ones [3]. In our solution for the NE-localization and value extraction tasks, we take another approach and *enhance* the manually created grammars to account for the misrecognitions typical for the employed ASR-system.

2. Three tasks of named entity processing

We distinguish three major subtasks in the named entity processing field:

2.1. Detection

The goal of named entity detection is to decide if utterances contain named entities of the specified type, while finding of the NE-instances themselves is not required. This question arises, for instance, when we try to exploit the presence of named entities in utterances to help calltype classification. This is justified by strong co-dependencies between occurrences of named entities and calltypes observed in many corpora [3]. Another possible implication of NE-detection is the modification of dialog manager behavior. Suppose, there are strong indications that the user just specified a date of some event, while the exact value of this date couldn't be reliably extracted. In this case, the system can reprompt, relating to the already given information: *"Please, repeat the date information once again"*, rather than concede the failure (*"I didn't understand you"*), followed by the previous prompt again. Other applications of the NE-detection for the dialog are conceivable [3].

2.2. Localization

In this task we strive to find out how many NE-instances are present in the utterance and to determine their exact locations. So, for applications like SCANMAIL [6] designed to provide a high-end textual interface to voice mail, it is essential to find out the exact location of important information in the ASR-transcription of the utterance and draw user's attention to it by highlighting the region or recovering its audio representation. There is no need for the system to actually *understand the meaning* of the found named entity, just localizing all its instances in the signal will suffice.

2.3. Value extraction

Finally, we can extract the values of all localized named entities. If we want to design a system that is capable of conducting dialog with the user by its own means, information extraction will become its indispensable component. For instance, in the

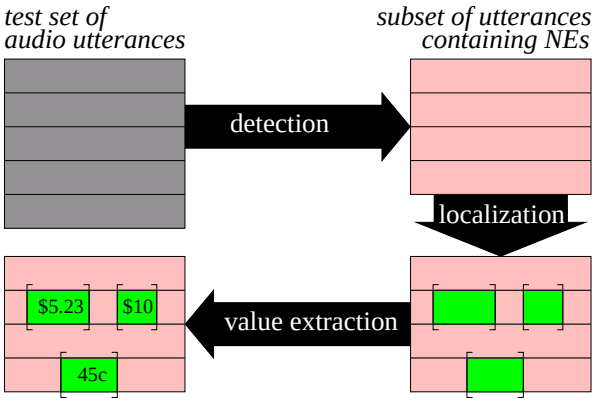

Figure 1: *Three stages of named entity processing.*

HMIHY task described in [7], there are numerous ways for the callers to communicate the date of a particular phone call. *"May six of the year two thousand"* and *"the sixth of May two thousand"* certainly mean the same (at least in the application context), albeit the user uses different linguistic means to convey this meaning. In order for the system to react adequately to a user's request, it must prescind from a particular wording and concentrate on the salient bits of information. In other words, a *normalization* has to be carried out. In our example, both phrases will be transformed into something like "05.06.2000".

These three subtasks can often be considered in a cascade connection: before we start looking for begin and end positions of NE-instances in the utterance, it is worthwhile to activate the detection mechanism first for an accurate prediction as to the presence of any named entities in the utterance at all. Given that issuing such a prediction is usually comparatively inexpensive, and the percentage of utterances containing named entities can be fairly low [3], the amount of saved computational power is considerable. Similarly, it is needful to localize an interval with a putative NE-instance, before trying to extract its value. This dependency is schematically illustrated in Figure 1.

3. Named entity detection

In this section we describe our classification-based approach to named entity detection. To make a prediction about presence of named entities in the utterance, we reformulate the task so as to look at it from the classification perspective: we introduce one class for each NE-type, as well as one "rejection" class that all utterances not containing NE-instances of any type are thrown into. Then, the detection task becomes a simple multi-class[1] classification task. Since indicators of named entity presence can be found throughout the entire utterance (cf. this example: *"I want to dispute the following charges on my last month bill: you made me pay* **twenty dollars** *for… "*), we decided to use LLAMA SVM-classifier to combine such (possibly weak) evidences from anywhere in the utterance transcripts. In particular, sequences of up to 5 words from the training corpus were employed as classification features, constituting a particular case of a positive definite symmetric rational kernel for classification on lattices [8]. This kernel was further enriched by a third degree polynomial transformation (polynomial cubic kernel in

[1]It is certainly possible for one utterance to contain named entities of different types.

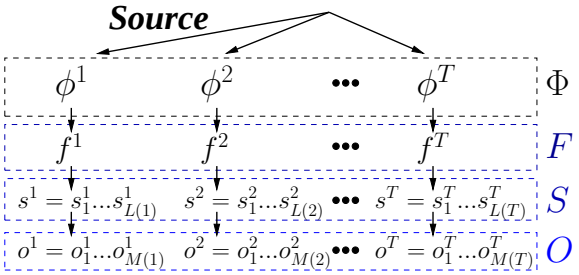

Figure 2: *Generative production mechanism.*

SVM). The classification with many classes is implemented in LLAMA by means of an optimal multi-class recombination of binary one-against-the-rest SVMs [9], and the final classification result is one *detection score* for each class. Should the achieved detection score of some class (not rejection) exceed a pre-specified threshold, the utterance is considered to contain at least one instance of the corresponding NE-type. If detection is part of the cascade model from Figure 1 this score can be passed on to the localization module.

4. Named entity localization

Unlike the detection stage, localization and value extraction require an explicite model of each NE-type we want to search for in the utterances. The discussion whether stochastic models or handcrafted grammars are better suitable for the NE-modeling purposes has its roots in the long-fought battle between empiricism and rationalism in speech recognition and understanding. While handcrafted grammars lead to superior results on clean data, such as written text, in the cases where no spelling information is available or noise (in form of misrecognition errors) is present, statistical approaches turned out to outperform rule-based systems [2, 10]. It is certainly much cheaper and faster to write a handcrafted grammar of high generalization power that accounts for many various NE-formulations than put together and manually transcribe and label a training corpus containing all these variants. Yet, the problem with the manual grammars is their deficient robustness towards misrecognition errors.

Methods of parallel combination of manually created and statistically estimated NE-grammar fragments were already presented in the literature [3]. In this paper, we report on our experiments towards enhancing the rule-based approach by incorporating the typical misrecognitions into the handcrafted grammars. While the ideal strategy would be to collect misrecognition statistics for each named entity expression separately, this again would require considerable labeling efforts and also suffer from the typical data sparseness. Our strategy is to separate the statistic methods of error compensation from the semantics of the task, the former being derived from the confusion matrix of a validation corpus, recognized with the same ASR, and the latter articulated in the handcrafted (application-dependent) named entity grammar fragments.

The idea behind our localization strategy is to find a maximum likelihood parse of the ASR-output in terms of named entity instances. The lexicon for parsing L consists of *fragments* ϕ_i of two kinds: the words $\{w_i\}$ present in the ASR-dictionary, and named entity grammar fragments, one for each considered named entity type.

To solve the parsing task, we employ the generative pro-

duction model from Figure 2 (see also [11]). In this sequential approach, the source emits a sequence Φ of fragments ϕ^t. For each fragment, one valid path f^t through it is realized (possibly with distortions) by some word sequence $s^t = s^t_1 \dots s^t_{L(t)}$ from the ASR-output. Each segment s^t also reflects an interval o^t of acoustic observations $o^t_1 \dots o^t_{M(t)}$ from the input audio signal O, and together these segments constitute a segmentation S of the ASR-output. With this generative model and some assumptions of mutual independency, we compute the likelihood of a parse as:

$$P(O,S,F,\Phi|L) \approx \underbrace{P(o^t|s^t)}_{\text{acoustics}} \times \underbrace{P(s^t|f^t)}_{\text{misrecognitions}} \times \underbrace{P(f^t|\phi^t)}_{\text{NE-grammar}} \times \underbrace{P(\phi^t|\phi^{t-1}_{t-K+1})}_{\text{language model}}, \quad (1)$$

a decomposition in four terms based on acoustic costs, misrecognition costs, intrinsic grammar costs and language model costs. Additionally, the detection scores can also be taken into account here. To obtain the sequences Φ^*, F^* and S^* that maximize (1), we employ the FSM-formalism and search for the best path $\mathcal{P}$ through the composition of three weighted finite state transducers:

$$\mathcal{P} = \text{bestpath}(\mathcal{GL} \circ \mathcal{D} \circ \mathcal{S}), \quad (2)$$

where $\mathcal{GL}$ represents the language model with the embedded grammar fragment FSMs, $\mathcal{D}$ is the distortion transducer and $\mathcal{S}$ encodes the ASR-output that, in general case, can also be a weighted word lattice. Special markers integrated in the output labels of the grammar fragments will indicate where in this parse each encountered named entity instance starts and where it ends.

To compute the distortion probability $P(s^t|f^t)$ from (1), we resort to the word *mapping* probabilities of the form $P(x \rightsquigarrow y)$ (read: the probability that word x is (mis)recognized as word y). Depending on whether x, y or none of the above is the empty symbol ε, one can talk about substitution, deletion or insertion probability of a particular word. Then, dynamic programming is used to obtain $P(s^t|f^t)$ via aggregation of the mapping probabilities along the most probable alignment of strings f^t and s^t.

In the simplest case of the *exact matching*, all mapping probabilities are determined by the Kronecker delta function ($P(x \rightsquigarrow y)$ is 1 if $x = y$ and 0 otherwise), and, as a result, so is $P(s^t|f^t)$. Alternatively, in order to avail ourselves of the *approximate matching* strategy, we can estimate the word mapping probabilities from two aligned representations of a validation corpus: manual transcriptions and bestpath-output of the employed recognizer. In (2) the word mapping probabilities are encoded in the distortion transducer $\mathcal{D}$ which, in our case where these probabilities are context-independent, is a trivial one-state flower FSM.

5. Named entity value extraction

Using finite state transducers to model named entity grammar fragments allows us to carry out the first stage of the named entity value extraction already during the localization step. This can be achieved by using the input labels of these transducers as well. In Figure 3 a transducer is depicted that back-translates all expressions of the form *"N dollars"*, with N being a spelled-out number ranging from 1 to 99, into a sequence of numbers and markers[2]. During the value extraction step, this sequence

[2]For clarity's sake, we do not single out the special case of *"one dollar"*.

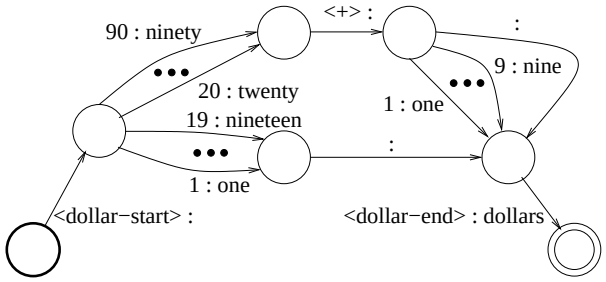

Figure 3: *Using transducers for named entity value extraction; for each arc, its input and output labels are separated by a colon.*

(taken from the input side of the resulting transducer $\mathcal{P}$) is post-processed to produce the normalized symbolic representation of the named entities that can be further forwarded to the understanding module. For example, the named entity instance *"twenty five dollars"* will be back-translated into *"<dollar-start> 20 <+> 5 <dollar-end>"*, which will then be parsed into the *meaning "$25"*.

One important advantage of using transducers to encode handcrafted named entity grammar fragments consists in the possibility of integrating semantic constraints in the syntactic models. For example, we can pay tribute to the fact that "02.29.2033" is an impossible date, by excluding it along with all its possible wordings from the grammar fragment, so that this date will be ignored already at the localization stage. Making a similar amendment for statistically trained named entity grammar fragments wouldn't be as easy.

6. Experiments and results

We conducted our experiments on the HMIHY-corpus [7] using ∼36K training and ∼9K test utterances, recognized with a word accuracy of 78%. Three NE-types were considered: PHONE_NUMBER, DATE and a context-dependent named entity ITEM_AMOUNT which referred only to monetary expressions on the customer's bill. Each of these NE-types was represented in the corpus by ca. 1K training and ca. 300 test instances.

We used the F-measure, popular in the information extraction community, to evaluate performance of our algorithms in all three NE-tasks. Let P be the precision and R the recall of an algorithm. Then, the F-measure is defined as:

$$F = 2PR/(P+R). \quad (3)$$

The plot in Figure 4 shows detection ROC-curves for all three named entities. We see that phone numbers are detected best (F-measure at the operating point 0.93), while the named entity DATE and the context-dependent ITEM_AMOUNT have F-measures of 0.83 and 0.81 respectively. We explain this result by the fact that longer named entities provide more cues as to their presence, while short ones (like the month "May") are often overlooked and especially difficult to detect when misrecognized by the ASR.

To assess the localization performance, we align manual transcriptions and ASR-output of the test corpus and map the discovered NE-instances back onto the original labeled corpus. We say that localization was successful if there is an intersection of the manually marked and the localized instances. To evaluate the results of value extraction, all named entities are normalized

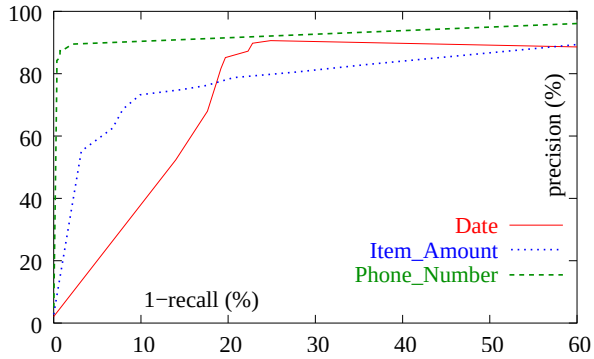

Figure 4: *Detection ROC-curves for three selected named entity types in HMIHY.*

named entity	baseline		prefilt. + approx.	
	local.	val. extr.	local.	val. extr.
ITEM_AMOUNT	0.61	0.57	0.67	0.64
DATE	0.74	0.70	0.75	0.71
PHONE_NUMBER	0.85	0.76	0.89	0.80

Table 1: *F-measure for localization and value extraction experiments.*

into symbol strings and the F-measure based on the Levenshtein distance between the reference and hypothesis strings is computed[3]. Suppose, the normalized reference of a phone number is "123.4567890" and the suggested hypothesis is ".1234067" (counting the area code delimiter). Having aligned these two, we will see that 6 symbols out of 11 are correctly identified, and 6 symbols out of the hypothesized 8 are correct. This produces: $R = 6/11, \quad P = 6/8, \quad F \approx 0.63$. Finally, for each named entity type the obtained F-measure values are averaged over all utterances in the test corpus.

Table 1 shows the results of NE-localization and value extraction experiments. The first experiment was a baseline where no upstream detection-based pre-filtering from Section 3 was done and only exact matching was allowed for ML-parsing on the localization stage. In the second experiment both listed extensions had been built in, and also a minimum context was introduced in the NE-grammars[4]. We see that the more sophisticated version of the algorithm resulted in a better localization and value extraction performance. Again, the best results were achieved for the long named entity PHONE_NUMBER. Furthermore, we observed that the detection-based pre-filtering reduced the number of utterances with attempted parses by a factor of 10.

Interestingly, increasing the order of the language model in (1) didn't improve the localization and value extraction significantly compared to the unigram case. We explain it by the fact that the manual NE-labels available for our training corpus also included a significant portion of immediate syntactic context for each named entity instance, whereas the grammar fragments actually used for the instantiation did not.

[3]For falsely localized instances F is always assumed zero.

[4]Modeling context turned out to be impractical with exact matching strategies.

7. Conclusions

We have shown how named entity processing can be subdivided into three subtasks: detection, localization and value extraction. We suggested methods for solving all of these subtasks and, in particular, explained how their cascade-based application facilitates an efficient algorithm for named entity value extraction. Evaluated on the HMIHY-corpus, our methods delivered very good results when using the SVM-classifier for the detection task. Furthermore, the advantage of the approximate matching for maximum likelihood parsing, when applied to localization and value extraction of rule-based named entity grammar fragments in speech, was demonstrated. We observed the best results for the named entity PHONE_NUMBER with an F-measure of 0.93, 0.89 and 0.80 for detection, localization and value extraction tasks respectively, and explained this result by a relatively high typical length of its instances.

8. References

[1] Appelt D., Hobbs J., Bear J., Israel D., Kameyama M., Kehler A., Martin D., Meyers K., Tyson M.: *"SRI International FASTUS System: MUC-6 Test Results and Analysis"*; in Proc. of MUC-6, pp.237–248, 1995.

[2] Bikel D., Schwartz R. and Weischedel R.: *"An Algorithm that Learns What's in a Name"*; in "Machine Learning", special Issue on Natural Language Learning, 34(1–3):211–231, 1999.

[3] Béchet F., Gorin A., Wright J. and Hakkani-Tur D.: *"Detecting and Extracting Named Entities from Spontaneous speech in a Mixed-initiative Spoken Dialogue Context: How May I Help You?"*, SpeechCom, 2003.

[4] Chinchor N.: *"MUC-7 Named Entity Task Definition"*; in Proc. of the MUC-7, 1997.

[5] Collins M. and Singer Y.: *"Unsupervised Models for Named Entity Classification"*; in Proc. of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp.100–110, University of Maryland, MD, 1999.

[6] Whittaker S. et al.: *"SCANMail: a Voicemail Interface that Makes Speech Browsable, Readable and Searchable"*; in Proc. of SIGCHI Conf. on Human Factors in Comp. Systems, Minneapolis, 2002.

[7] Gorin A., Riccardi G., Wright J.: *"How may I help you?"*; SpeechCom, 23:113–127, 1997.

[8] Cortes, C., Haffner, P. and Mohri, M.: *Rational Kernels*; in Advances in Neural Information Processing Systems, 15, March 2003.

[9] Haffner, P., Tur G., Wright, J.: *Optimizing SVMs for Complex Call Classification*; in Proc. of ICASSP, Hong Kong, April 2003.

[10] Palmer D., Ostendorf M. and Burger J.: *"Robust Information Extraction from Spoken Language Data"*; in Proc. of Eurospeech, pp.1035–1038, Budapest, Hungary, 1999.

[11] Levit M., Gorin A. and Nöth E.: *"Using EM-trained String-Edit Distances for Approximate Matching of Acoustic Morphemes"*; in Proc. of ICSLP, pp.1157–1160, Denver, Colorado, 2002.