Road Segment Identification in Natural Language Text

Ahmed Y. Tawfik and Lawrence Barsanti

School of Computer Science, University of Windsor
Windsor, Ontario, Canada
Email atawfik@uwindsor.ca, barsant@uwindsor.ca

Abstract. This paper describes a technique to extract geographic location information from a natural language description of a location. The technique relies on a set of domain specific tags and a set of keywords. The tags are used to identify roads, intersections, and landmarks. Tag combinations are used to discover road segments. The technique is applied to understanding highway construction reports for the Canadian Province of Ontario.

1. Introduction

Location information has traditionally been expressed in two main forms: natural languages and maps. Maps represent a rich visual representation that captures a host of spatial relationships among collocated elements. Natural languages provide a set of focused abstractions of the spatial relationships represented in a map. In natural languages, the choice of the relevant abstraction is generally task dependent. Maps and natural language interfaces to geographic information systems continue to be complementary. For example, systems that generate driving directions like Yahoo! Maps, MapPoint and MapQuest [6] provide a linguistic description of a map. Coral [1] applies natural language generation techniques to make the linguistic description more natural. Understanding and visualizing textual geographic references on a map has attracted less attention as a research focus. By grounding named entities to spatial locations, a system can answer spatial queries [3]. A geo-parser combines data from multilingual gazetteer with natural language text and a geographic information system to produce a map highlighting the locations mentioned in the text [4].

The focus here is on defining a set of special purpose tags that are designed to understand urban location descriptions like driving directions that can be used in translating location information expressed in natural language to a segment or region on a map. The application that has motivated this work is building a system that determines the location of highway construction based on construction report summaries. These summaries include some structured fields (e.g. affected highway, closest city, length of construction) and a natural language description of the traffic impact. The traffic impact typically includes detailed location information. Figure 1 shows an example of highway construction summary for highway 401 in Ontario[1], Canada.

[1] From the Ontario highway construction reports available at http://www.mto.gov.on.ca/english/index.html

Start of Construction:	June 01, 2004
Estimated End of Construction:	November 25, 2005
Highway:	401
Length of Construction:	10.6 kilometers
Close To:	Tilbury
Type of Contract:	Road Construction
Traffic Impact: Highway 401, from Highway 77 easterly To Essex County Road 42. Highway 401 will be reduced to a single lane of traffic in each direction separated by temporary concrete barrier wall. The speed limit is reduced to 80 kilometers per hour.	
Region of Ontario:	Southwestern

Figure 1. Sample Construction Report Summary

Section 2 presents the knowledge representation that serves as a foundation to the work. Section 3 introduces the two level parsing technique used in the interpretation of some natural language location descriptions. Section 4 presents the results of analysing construction reports and Section 5 presents a brief conclusion.

2. Elements of the Knowledge Representation

Topological and metric spatial relationship expressed in natural language has to be interpreted before the location can be correctly determined. In general, we consider that we have linear entities and regions. A road is represented as a linear entity. Linear entities include highways, creeks, rivers, and boundary lines. Towns, cities, counties, and mountains are considered as regions. The intersections of two lines define a point. The intersection of a line and a region defines a line segment. Specifying a location relies on the identification of the relationship that holds between lines or between a line and region. Interpreting the natural language terms describing these relationships relies on the two-level part-of-speech-tagging described in the next section.

The knowledge representation is based the 9-intersection model [5]. According to this model, each spatial object divides the space into three components: the boundary of the object, the space internal to the object, and every thing else is external to the object. Therefore, a simple line (that has no self loops) has two boundary points, and a continuous sequence of internal points joining the two boundary points. Similarly, a region has a closed boundary, an internal area and an external area. For simplicity, we assume that the map is a 2D space.

Line-line Relations

Shariff et al. [5] identify 33 topological relationships that may hold between two lines. To simplify the representation, we omit self-similar (symmetric) relationships. A relationship is self-similar if its inverse has the same 9-intersection matrix as the

original relationship The 33 relations include 11 self-similar relationships in addition to 11 relationships with 11 respective inverses. . For example, ***equal*** (LL22) and ***intersect*** (LL2) are self-similar relationships. However, ***contains*** (LL5) has an inverse ($LL5^{-1}$).

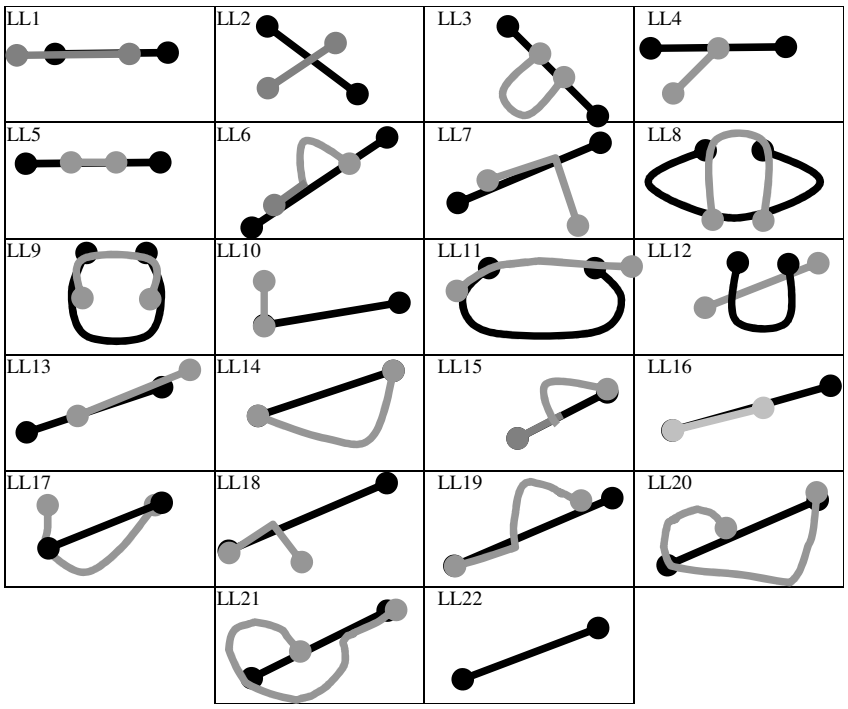

Figure 2. Line-Line Relationships

Figure 2 illustrates the topological relationships that may hold between two lines. It is possible to divide these topological relationships into relationships involving overlapping segments (LL5, LL6, LL7, LL11, LL12, LL15, LL16, LL18, LL19, LL21, and LL22) and others involving 0, 1 or 2 boundary points (LL1, LL2, LL3, LL4, LL8, LL9, LL10, LL13, LL14, LL17, and LL20).

3. Parsing Location Information

In order to understand location text, it is important to extract references to locations in the text. In general, this a difficult problem as special attention should be given to the use of prepositions (at, from, to, …etc.) and a great deal of disambiguation may be involved in distinguishing references to places from other proper nouns. A gazetteer is useful in distinguishing references to cities and towns from other proper nouns in the text. However, some commonsense knowledge is necessary to correctly parse “Mr.

England flew to India" and figure out that England refers to a person while India is a place. Fortunately, in our application, location information was easily identifiable and a rather limited amount of effort went into disambiguation. The tagging technique introduced here assigns special tags to spatial words and phrases indicating direction as in northerly, left, southbound, and Windsor-bound. Special tags are also assigned to words and phrases denoting proximity or distance like near, close, next to, or distance (like a mile). As some numbered highways also have a name (e.g. County Road 19 is also Manning road), a special tag (ALT_ROAD) is necessary. Table 1 lists the set of domain specific tags used here.

Table 1. Domain Specific Tags

Tag	Represents
INT_ID	Intersection Identifier
ROAD	Road / Highway
OFF	Offset/Proximity/Distance
DIR	Direction of Traffic (set of lanes)
NLM	Natural Landmark
MLM	Manmade Landmark
ALT_ROAD	Alternate Road Name

The assignment of these domain-specific tags is performed as a second level tagging after the text has been tagged using a standard part-of-speech tagger. Here, we use CLAWS [2] for first level tagging. CLAWS tag set includes locative tags NNL, NNL1 and NNL2). However, we found that the names of most roads and landmarks consist of a sequence of singular common nouns (NN1), singular proper nouns (NP1), numbers (MC), and in some cases title nouns (NNSB). Phrases that contain these tag sequences are of interest, we identify these phrases as potential name phrase (PNP). A PNP is defined as a sequence of one or more words whose tags are any combination of the NN1, NP1, and NNB tags; see Figure 3. Locative tags still play an important role in identifying locations. For example, both roads and natural landmarks can be found by searching for the sequence of tags PNP NNL1. It is the word represented by the NNL1 tag that distinguishes between them. That is why some of the tag sequences presented in Table 3 have keywords associated with them. Table 2 lists the keywords used in assigning spatial tags. Adjectives (JJ), prepositions (II), nouns of direction (ND), units of measurements (NNU), the preposition "for" (IF), and participle (or past) form of verbs (VVN/VVD) all proved useful in identifying road segments.

Text	Highway	77	,	From	Highway	77	Easterly	To	Essex	County	Road	42	.
Tags	NN1	MC	,	II	NN1	MC	JJ	II	NP1	NN1	NN1	MC	.
	PNP				PNP				PNP				

Figure 3. Example of Potential Name Phrase Tags

Notice that in Table 3 some of the patterns contain domain specific tags. For this reason the order in which tags are found is important; the following order is used: INT_ID, PNP, ROAD, ALT_ROAD, OFF, DIR, NLM, MLM.

Table 2. Keyword Lists

Feature	Keywords
Road Indicator	avenue, boulevard, parkway, way, expressway, drive, road
Numbered Roads	Highway, route, road
Natural landmark	river, creek, brook, lake, island, isle, islet, narrows, mountain, forest
Manmade landmark	bridge, span, overpass, underpass, tunnel, structure, culvert, skyway
Direction	Northbound, northward, southbound, southward, eastbound, eastward, westbound, westward
Destination	Bound
Intersection	Intersection, junction, crossroad, crossway, crossing, corner, interchange
Road type	regional, municipal, county
Directional Adjective	Northerly, southerly, easterly, westerly

Table 3. Patterns for detecting domain specific tags

Special Tag	Sequence	Keywords From Table 2	Example
INT_ID		Intersection	
ROAD	PNP MC	PNP ends with Numbered Rd	County Road 42
	PNP NNL1	NNL1 not a landmark	Queen Street
	PNP NNL1 MC	NNL1 not a landmark	County Road 121
	JJ PNP MC	JJ Road Type	Regional Road 3
	PNP	Starts with a numbered road or ends with a road indicator	Highway QEW Van Horne Ave.
OFF	ND1 IO		East of
	MC NNU1 ND1 IO		1 kilometer East of
	MC NNU2 ND1 IO		2 kms East of
	JJ MC NNU1	JJ directional adjective	Northerly 1.0 km
	JJ MC NNU2	JJ directional adjective	northerly 2.3 kms
	JJ IF MC NNU1	JJ directional adjective	Northerly for 1 km
	JJ IF MC NNU2	JJ directional adjective	Northerly for 2 kms
DIR	ROAD ANY	ANY is direction	Highway 401 westbound
	ANY ROAD	ANY is direction	Eastbound Highway QEW
	PNP VVD	VVD is destination	Toronto Bound
	ND1 VVN	VVN is destination	west bound
	ANY	ANY is direction	Westbound
	ROAD ND1		Highway 8 East
NLM	PNP	Last word natural landmark	Pike Creek
	PNP NNL1	NNL1 in natural landmark	
MLM	NLM ANY	Any in manmade landmark	Pike Creek Bridge
	PNP	Last word in manmade landmark	Thorold Tunnel
ALT_ROAD	(ROAD)		(Manning Road)

However, using only patterns does not provide good enough results. For instance some people may use short forms like "Take Highway 401 to Walker Road. Get off the 401 and go south on Walker." In this sentence both "the 401" and "Walker" refer to roads, but would not be matched by any of the listed patterns. To handle this type of situation every time a road is found the rules in Table 4 are used to generate potential short form for the road. After all the patterns have been searched a second pass can find and tag these short forms.

Lastly, in some cases CLAWS tags a word like road as a noun (NN1) when it should be a locative noun (NNL1). That is why there are redundant rules like "proper noun followed by a number" (PNP MC) and "proper noun followed by a common noun and a number" (PNP NNL1 MC).

Identifying Road Segments

Usually, road segments are defined as either the stretch of road between two points (Figure 4a), or as a stretch of a given length starting at a given point (Figure 4b). For example, construction may affect a highway segment between two intersections or it may affect the area around a bridge or intersection. According to the 9 intersection model, a point is the intersection of two lines (LL2) or a segment and a line (LL4). A segment can also be defined in terms of the portion of a line intersecting a region. In this case, the points defining the ends of the segment are defined by the intersections of the region boundaries with the line.

Table 4. Rules for creating short forms

Pattern	Short Form
PNP MC	"the MC" and "the highway" (if PNP is the word highway)
PNP NNL1	PNP (i.e. road name without street, road, etc…)
PNP NNL1 MC	"the MC"
JJ PNP MC	"the MC"
PNP	"the PNP" without first word or PNP without last word

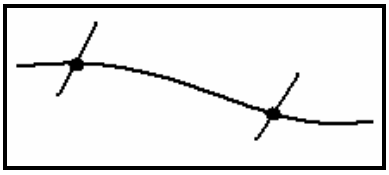

Figure 4a. Segment delimited by two points

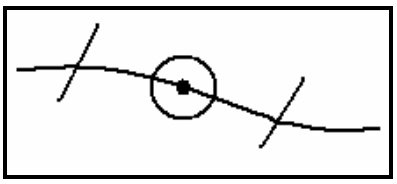

Figure 4b. Segment around a point

Table 5. Segment Identification Rewriting Rules

Type	Pattern	Rewrite As
Segment with two points	ROAD[1] ROAD[2] to ROAD[3]	ROAD[1] & ROAD[2] to ROAD[1] & ROAD[3]
	ROAD[1] ROAD[2] to NLM	ROAD[1] & ROAD[2] to ROAD[1] & NLM
	ROAD[1] NLM to ROAD[3]	ROAD[1] & NLM to ROAD[1] & ROAD[2]
	ROAD[1] NLM[1] to NLM[2]	ROAD[1] & ROAD[2] to ROAD[1] & MLM
	ROAD[1] ROAD[2] to ROAD[3]	ROAD[1] & MLM to ROAD[1] & ROAD[2]
	ROAD[1] between ROAD[2] and ROAD[3]	ROAD[1] & ROAD[2] to ROAD[1] & ROAD[3]
	Note: if DIR was found using either the 1st or 2nd pattern in Table 3 it could substitute for one of the ROAD tags. i.e. ROAD westbound ROAD to MLM	
Segment with a single point	ROAD[1] at ROAD[2]	ROAD[1] at ROAD[2] (no interjecting words)
	ROAD at NLM	ROAD at NLM (no interjecting words)
	ROAD at MLM	ROAD at MLM (no interjecting words)
	ROAD[1] OFF[1] ROAD[2] OFF[2]	ROAD[1] at ROAD[2]
	ROAD[1] ROAD[2]	ROAD[1] at ROAD[2]
	ROAD NLM	ROAD at NLM
	ROAD MLM	ROAD at MLM

Table 5 shows how the domain specific tags and CLAWS part-of-speech tags are used to identify road segments. The sequences used to identify road segments consist of tags and keywords. The tags and keywords in a pattern have to appear in the order laid out by the pattern; but they do not have to be consecutive. For example, the sentence in figure 5 matches the first pattern in Table 5 and produces the phrase "Highway 58 & Regional Road 3 to Highway 58 & Forks Road". The rewriting rules in Table 5 try to deal with implicit references to points and intersections. For example, in the phrase "Construction on Highway 22 from Howard Ave to Walker Rd." implicitly means that the construction starts at the intersection of Howard Ave. and Highway 22 and ends at the intersection of Walker Road and Highway 22.

Text	Highway	58	,	Regional	Road	3	To	South	Of	Forks	Road	.
Tags	ROAD		,	ROAD			II	OFF		ROAD		.

Figure 5. Example of Road Segment Identification

4. Implementation and Performance Results

The algorithm described in this paper has been implemented in Java. The implementation consists of two main components: a special purpose tagger and a road segment identification module. It uses a separate file for the keywords and patterns shown in Tables 2, 3, 4, and 5 to maintain modularity and make it easy to add new patterns or keywords. An input text is first submitted to CLAWS to get a part-of-speech (POS) tagged text. In the POS tagged sequence, keywords are identified and the sequence is matched to the patterns given in the keywords/patterns file to get a text tagged with our application-specific tags in addition the part-of-speech tags. If a word could have more than one POS tag, the tags are considered in the order of their likelihood until a matching pattern is found. If a word sequence matches multiple patterns, (e.g. PNP NNL1 and PNP NNL1 MC), the longest sequence, that overlaps the shorter one, is used. The road segment identification module applies the rewriting rules in Table 5 and generates a list of road segments. The algorithm looks for road segments delimited by two points first before trying to identify road segments that are near a point. The following example illustrates the outputs from the tagger and the segment identifier.

Input	*Hwy 401* , 10 Kms east of Interchange Number 661 *At* the *Donovan Creek Bridge*													
POS	NN1	MC	,	MC	NNU2	ND1	IO	NN1	NN1	MC	II	AT	NP1	NN1	NN1
Domain Tag	ROAD			OFF				INT_ID					MLM		
Segment	(ROAD AT MLM) OFF: Hwy 401 at Donovan Creek Bridge, 10 Kms East														

To test the performance of the tagger, we used 25 construction reports from Ontario road construction web site. Each construction report was manually tagged using the domain specific tag set, and then the reports were tagged by the tagger. The tags found by the domain tagger were then checked for incorrect tagging. The results are summarized in Table 6.

From the results it clear that the patterns for both natural and manmade landmarks contributed a large number of errors. There are too many false positives for natural landmarks. A more semantic approach to finding natural landmarks should reduce the number of false positives because the false positives all occurred when the tagger failed to find either a manmade landmark (i.e. the rule NLM ANY failed because of the keyword list) or road. As for manmade landmarks, a wider range of patterns and better generalization would probably improve their results as the test set simply contained keywords that were not identified in the initial analysis.

To evaluate the performance of the road segment identification, twenty-three of the twenty-five construction reports used for domain tagging were read by a person who looked for occurrences of road segments as explained in Figure 4. The tagged reports were processed by the system and the strings it produced were reviewed for correctness and counted. Additionally, all of the misses and partially correct (e.g. near A instead of between A & B) results were examined to determine if the problem was a consequence of the domain tagging. Table 7 presents a summary of the results.

We found that inaccuracies in the tagging seriously degraded the quality of the information extracted. A second look at the tagging results revealed that even though

the domain tagger has an average accuracy of 76.7% only 12 of the 25 reports (48%) were tagged flawlessly. In fact, every other report had at least one error, which could easily throw off the segment identification.

Table 6. Tagging Results

Tag	Actual	Total Found	False Positives	Percent Correct
INT_ID	12	12	0	100
ROAD	63	50	2	76.2
OFF	28	26	1	89.3
DIR	17	13	0	76.5
NLM	1	4	4	0
MLM	14	7	1	42.9
ALT_ROAD	2	1	0	50
Weighted Average				76.7

Table 7. Road segment identification results

Type	Actual	Correctly Identified with automatic tagging	Correctly Identified with manual tagging
Two Points	10	3	9
Single Point	19	12	16
Other	1	0	0
Overall	30	15	25

We then added a CITY tag that matches the city name from the structured information associated with the construction report. The CITY tag is useful to avoid interpreting a phrase like “Riverside Drive, Windsor” as an intersection.

Surprisingly, in our tests of the road segment identification algorithm, no false positives were produced; however in some cases weaker, but correct information was found (e.g. near A instead of between A & B). This is surprising because the last three patterns in Table 5 are rather lenient. Also, note that information such as direction and offset could also be leveraged in order to improve the extracted information. For instance, the pattern ROAD1 OFF1 ROAD2 OFF2, occurred in four reports and was defined as a region around an intersection, but it would have been better defined as a road segment delimited by two points.

5. Discussion and Conclusions

Determining the location of the highway construction described in Figure 1 requires parsing the text to extract the starting landmark or intersection (Highway 401 and Highway 77) and the ending landmark or intersection (Highway 401 and Essex County Road 42). Using a gazetteer as a dictionary to look up road names, landmarks, and populated places should improve the performance [4]. However, in many cases the construction zone is bound by harder to define landmarks. Consider the following examples:

- "Highway 35, Victoria/Haliburton Boundary Northerly for 8.1 kilometres to 0.3 kilometres north of Miners Bay."
- "Highway 21, from the north limits of Goderich northerly for 2.0 kilometres and Straughn's Creek Culvert 8 kilometres south of the town of Goderich."
- "Highway 401, from 2.55 kilometres west of Boundary Road, easterly to 0.75 kilometres east of Boundary Road and westbound lanes 0.5 kilometres east of Brookdale Avenue, easterly for 0.5 kilometres."

In the first example, the construction zone is defined in terms of the highway intersection with the boundary between two regions. In the second example, the construction zone is apparently discontinuous as it spans two kilometres from the north limits of a town and 8 kilometres from a small creek to the south of the same town. In the third example, the word "Boundary" is the name of a road not a region boundary as in the first example.

This work has provided a technique to identify road segments that are of interest for some reason (in our case, they were affected by construction). The technique can be useful in other applications like understanding driving directions. The results reported here are for a relatively small test corpus obtained from a single source and may not be statistically significant. However, these results highlight some of the strengths and limitations of the proposed approach.

Acknowledgements

This work is supported by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC), Canada.

References

1. Dale, R., Geldof, S., and Prost, J.-P.: CORAL: Using Natural Language Generation for Navigational Assistance", Twenty-sixth Australasian Computer Science Conference (ACSC2003), Adelaide, South Australia, (2003).
2. Garside, R., and Smith, N. : A Hybrid Grammatical Tagger: CLAWS4, in Garside, R., Leech, G., and McEnery, A. (eds.) Corpus Annotation: Linguistic Information from Computer Text Corpora. Longman, London, (1997) 102-121.
3. Leinder, J., Sinclair, G., Webber, B.: Grounding Spatial Named Entities for Information Extraction and Question Answering. Workshop on the Analysis of Geographic References at NAACL-HLT 2003 Conference, Edmonton, Alberta, Canada, (2003).
4. Poluiquen, B., Steinberger, R., Ignat, C. and De Groeve, T.: Geographical Information Recognition and Visualization in Texts Written in Various Languages. The 2004 ACM Symposium on Applied Computing (SAC'04), Nicosia, Cyprus, (2004) 1051-1058.
5. Shariff, R.B.M., Egenhofer, M.J., Mark, D.M.: Natural-Language Spatial Relations between Linear and Areal Objects: The Topology and Metric of English-Language Terms. International Journal of Geographical Information Science, Vol. 12 No. 3, (1998) 215-246.
6. Forth, S.: Online Mapping Services Guide the Way. Plugged In, 16 :11 (2005) 48-50.

Learning Discourse-new References in Portuguese Texts

Sandra Collovini and Renata Vieira

Universidade do Vale do Rio dos Sinos
CP 275, CEP 93022-000, São Leopoldo, RS, Brazil
sandrac@exatas.unisinos.br,renatav@unisinos.br

Abstract. This work presents the evaluation of a discourse status classifier for the Portuguese language. It considers two distinguished classes of discourse novelty: *Brand-new* and *New* references. An evaluation of the relevant features according to different linguistic levels are presented in detail.

1 Introduction

The identification of discourse status has been recognized as a relevant task in natural language understanding. Many systems have been proposed to classify referring expressions [2, 14, 12, 5, 13, 8] in order to recognize if they are new or old information. This comes along with the problem of anaphora resolution, it is usually useful establish the relations of old expressions with their antecedents. It is important, for instance, to identify antecedents for pronouns (it, he, she) to interpret the meaning of the discourse. Our work focuses on definite descriptions (DDs), those referring to expressions with a definite article (such as *the boy, the girl*) , because they are numerous in texts and are the main source of ambiguity regarding novelty, as opposed to other expressions. Pronouns, for instance are mainly old and indefinite descriptions are mainly new.

Whereas most of the literature in this area refers to the English language, we built and evaluated a system to classify discourse status in Portuguese texts. Besides proposing and evaluating such a system for a new language, this work is original by considering two different classes of discourse-new DDs. At first, we classified *Brand-new* definite descriptions. However, as the distinction between *Brand-new* and *Anchored-new* DDs is remarkably difficult [9], a second study was made considering the more general class *New*, which includes both *Brand-new* and *Anchored-new*. Also original in this study is that the relevance of the features used for learning the classifier is analyzed considering different levels of linguistic knowledge.

This paper is organized as follows. In Section 2 we discuss the related work. Classes of discourse status are defined and exemplified in Section 3. In Section 4, a corpus study and the features used to build our classifier are presented. In Section 5 we discuss the resulting decision trees and the relevance of the features is discussed in Section 6. In Section 7, this work also shows an evaluation of the resulting system on completely unseen data. In Section 8 we present our final remarks.

2 Related Work

There are in the literature several proposals of referring expressions classifiers. In [2] a classifier for DDs was developed. The authors define new DDs as independent existential expressions, understood by the readers isolately, without needing a context. This system conjugates 9 syntactic heuristics (restrictive pre-modifiers and post-modifiers, relative clauses, adjective constructions etc.) and other heuristics like DDs that occur in the first sentence of a text. As a result, they achieved 78% of recall, 87% of precision and 82% of F-measure for the classification of independent existing DDs. In [14] a heuristic based discourse-new DD classification system was developed, reaching 69% of recall, 72% of precision and 70% of F-measure, on the basis of 9 such features.

In [13] a classifier for discourse-new DDs and unique expressions was presented, where discourse new DDs were defined as the first mention of an entity in the discourse and unique expressions were said to specify their referent totally, and for this reason are understood without any context. The author took into consideration 32 features (syntactic, contextual and definite probability) including data from the web. The reported result was 82.3% of recall, 84.8% of precision and 83.5% of F-measure in discourse-new DDs classification and 68.8% of recall, 85.2% of precision and 76.1% of F-measure were reported for unique expressions classification. In [8] a group of common features in these previous work for another discourse-new DDs classifier (9 features) was reviewed and applied. The classifier resulted in 95.1% of recall, 85.8% of precision and 90.2% of F-measure.

All works cited above refer to the English language. Some other languages are also studied but not so extensively [1, 4, 5]. There are some corpora studies about coreference for the Portuguese language [11], but to the best of our knowledge there is no implemented DD resolution or classification system for Portuguese, so far. In addition to that, another difference of our work is that we give a detailed analysis of the features that were actually relevant to the classification, whereas in these previous work there is usually none. An exception is [7], which examines anaphoricity information to improve a learning-based coreference system and presents a list of the most informative features.

This work is the first one to present a classifier for DDs in Portuguese language. Based on a corpus study, DDs were analyzed and a set of features was organized in 3 groups considering three distinct linguistic levels. Features specifically related to the noun phrase structure constitute the first group, features which consider the sentence structure are in the second one, and the third group is based on information about the previous sentences. In the next section, we present the classes in detail.

3 Classes Description

The classes of DDs considered in this work are mainly based on [10], but they are also related to many of the studies discussed in Section 2. In the examples below, DDs are presented in boldface and their antecedents are underlined.

New DDs: these are definite referring expressions that introduce new entities into the discourse. In this work we consider two types of New DDs, *Brand-new* and *Anchored-new*.

- **Brand-New DDs** (discourse-new or non-anaphoric): introduce entities which are new in the discourse:
  A Folha de São Paulo *apresentou as listas apreendidas na operação contra o crime organizado.* ***[The Folha de São Paulo*** *presented the lists arrested in the operation against the organized crime.]*
- **Anchored-new DDs** (associative anaphors or bridging): refer to entities that have a semantic connection with an antecedent expression, which is necessary to their interpretation:
  A Folha de São Paulo apresentou as listas apreendidas na operação contra o crime organizado. O jornal tentou ouvir ***o delegado encarregado****. [The Folha de São Paulo presented the lists arrested in the operation against the organized crime. The newspaper tried to listen to* ***the police chief in charge.]***

Old DDs: refer to entities mentioned in the previous discourse. Old DDs can be *Plain-old* and *Related-old*.

- **Plain-old DDs** (direct anaphors): have an identity relation with their antecedents and share with them the same head-noun:
  *... as listas apreendidas na operação contra o crime organizado. Alguns delegados também são citados n****as listas****. ...* ***[the lists arrested in the operation against the organized crime****. Some police chiefs are also mentioned in the lists.]*
- **Related-old DDs** (indirect anaphors): have an identity relation with their antecedents however they present a distinguished head-noun:
  A Folha de São Paulo apresentou as listas apreendidas ... ***O jornal*** *tentou ouvir .... [The Folha de São Paulo presented the lists arrested ...* ***The newspaper*** *tried ...]*

4 Corpus study

Our work was based on two corpora. Corpus 1 was formed by 24 newspaper articles from Folha de São Paulo, written in Brazilian Portuguese, corresponding to part of the NILC[1] corpus. Out of 2319 noun phrases (NPs) we identified 1331 DDs. Corpus 1 was used for the learning phase. Corpus 2 was composed by 4 texts from the Public newspaper, written in European Portuguese from CETEMPublico[2] corpus. Out of 770 noun phrases we identified 482 DDs. Corpus 2 was used for the final evaluation.

The corpora were automatically annotated with syntactic information using the parser PALAVRAS[3] [3] to Portuguese. They were also manually annotated with coreference using MMAX [6]. The first annotation task was to distinguish *New* and *Old* DDs. The second task was pointing to the antecedent for the old cases. Corpus 1 was

[1] http://www.nilc.icmc.usp.br
[2] http://www.linguateca.pt/CETEMPublico
[3] http://visl.sdu.dk/visl/pt/parsing/automatic

annotated by three annotators. The agreement for the first task was close to 90.0%. Corpus 2 was annotated by four annotators. For the first task, the agreement resulted in 94.7% among the four annotators, for all other cases there was agreement among three annotators. This two-fold distinction is much easier than for the four classes, which explains why agreement was high whereas other work usually report much less than that. For the second task, antecedents annotation for those classified as old, four annotators agreed in 73.9% of the cases, in other 6.3% of the cases there was agreement among three annotators, in 0.84% only two annotators agreed, and complete disagreement was verified for the remaining 18.9%. The results are shown in Table 1.

Table 1: Manual Annotation

Corpus	**New DDs** (%)	**Old DDs** (%)	**Total** (%)
1	816 (61.3%)	515 (38.7%)	1331 (100%)
2	308 (63.9%)	174 (36.1%)	482 (100%)

The corpus was further analyzed, dividing *New* and *Old* DDs in their subclasses as presented in Section 1 (see Table 2). The usual large quantity of *Brand-new* DDs was confirmed. In Corpus 1, 52.3% were *Brand-new* and in Corpus 2 this number was even higher, 59.5%. DDs of Corpus 1 were studied against the features described in previous work, as presented in Section 3. A total of 16 features were identified in three groups of features according to different levels of linguistic knowlege. Group G1 considers information about the noun phrase alone, G2 considers information about the sentence in which the DD appears, G3 takes into account information about the previous text detailed in Table 3. Examples from the corpus illustrating each of the features are presented.

– PP: *Os membros da classe jurídica. [The members of the juridical class.]*
– APP: *O Prefeito de Gravataí, Daniel Luiz Bordignom. [The Gravataí major, Daniel Luiz Bordignom.]*
– PN_APP: *O delegado Elson Campelo. [The Police Chief Elson Campelo.]*
– REL_CL: *O texto que deve ser assinado pelos jornalistas. [The text that must be signed by jornalists.]*
– CPN_HEAD: *O Othon Palace Hotel. [The Othon Palace Hotel.]*
– AP: *As conversas mais antigas. [The older conversations.]*
– ADJ_PRE: *O primeiro grau. [The first degree.]*
– NUM_PRE: *Os 65 anos. [The 65 years.]*
– NUM: *Os anos 60. [The sixties (decade).]*
– PRON_DET: *Os nossos arqueólogos. [The (our) archaeologists.]*
– SUP_PRE: *Os melhores alunos. [The best students.]*
– SUP: *O Christofle líquido é o melhor. [The Liquid Christofle is the best.]*
– SIZE: *O quilômetro 430 da rodovia Assis Chateau Briand. [The 430 Km from Assis Chateau Briand road.]*
– COP: *O coreano seria a língua dos anjos. [(The) Koren would be the angels tongue.]*

These features were used for decision trees learning on the basis of examples from Corpus 1. After the learning process, the best resulting trees were implemented and further tested on unseen data (Corpus 2).

Table 2: New and Old subclasses

Corpus	**New DDs**		**Old DDs**	
	B-new (%)	A-new (%)	P-old (%)	R-old (%)
1	696 (52.3%)	120 (9.0%)	364 (27.35%)	151 (11.3%)
2	287 (59.5%)	21 (4.4%)	159 (33.0%)	15 (3.1%)

Table 3: Groups of Features

Groups	**Feature**	**Description**
G1	PP	Prepositional phrase.
	APP	Apposition.
	PN_APP	Appositive proper name with no explicit mark.
	REL_CL	Relative clause.
	CPN_H	When the head is a compound proper name.
	AP	Adjectival phrases.
	ADJ_PRE	Adjective preceding the head.
	NUM_PRE	Number before the head.
	NUM	Number after the head.
	PRON_DET	Other determinant besides the definite article.
	SUP_PRE	Superlative premodifier.
	SUP	Superlative alone.
	SIZE	Containing five terms or more.
G2	COP	DDs in a copular construction.
	S1	DDs that occur in the first sentence of the text.
G3	NO_ANT	DDs head is a word that does not occur previously in the text.

In [8] a set of 9 features from 6 groups (anaphora, predicative NPs, proper names, functionality, establishing relative, text position) was proposed. Our study takes 3 groups of features which are different from those presented in [8], but the features themselves are similar. They consider proper name, apposition, prepositional phrase, relative clause, superlative, copular construction, position in text, and anaphora. Our choice of 3 groups was motivated by the analysis of the NP alone, the NP plus sentence structure and position, and the NP, sentence plus previous text.

5 Decision Trees Learning

The learning algorithm used was Weka[4] *j48*, with 10 fold cross-validation. We tested different combinations of the 3 group of features for the decision trees generation: G1, G12 (=G1+G2) and G123 (=G1+G2+G3). Group G1 considers the noun phrase alone, G12 considers the noun phrase features and also information about the sentence, G123 will take into account noun phrase and sentence information but also the existence of a noun phrase with the same head as the DD in the previous text.

The first classification experiment considered the classes *Brand-new* (expressions that do not have an antecedent) and *Other* (expressions that have an antecedent). The results are presented in Table 4 and the features considered for the resulting trees in Table 5, in order of appearance in the trees. G123 presented the best results of precision, recall and F-measure for the *Brand-new* class, and the higher number of correctly classified occurrences in general. G1 alone, however, results in precision as high as other groups. It is in recall that G123 shows improvements when compared to the others. The number of features went down to 4 in G123.

Table 4: Brand-new classification
Correct(C); Precision (P); Recall (R); F-measure (F)

Classes	**G1**				**G12**				**G123**			
	C	**P**	**R**	**F**	**C**	**P**	**R**	**F**	**C**	**P**	**R**	**F**
B-new	63%	65%	55%	60%	64%	66%	57%	61%	70%	65%	88%	75%
Other		61%	70%	65%		62%	71%	60%		82%	53%	64%

Table 5: Features for classifying Brand-new DDs

Relevant features	
G1	SIZE, AP, CPN_H, ADJ_PRE,NUM_PRE, PN_APP
G12	S1, SIZE, AP, ADJ_PRE,CPN_H, PN_APP
G123	S1, NO_ANT, SUP_PRE, NUM

Table 6: New classification
Correct(C); Precision (P); Recall (R); F-measure (F)

Classes	**G1**				**G12**				**G123**			
	C	**P**	**R**	**F**	**C**	**P**	**R**	**F**	**C**	**P**	**R**	**F**
New	61%	71%	58%	64%	61%	71%	61%	66%	77%	76%	89%	82%
Other		53%	66%	59%		55%	66%	60%		81%	60%	69%

[4] http://www.cs.waikato.ac.nz/ml/weka

Table 7: Features for classifying New DDs

Relevant features	
G1	SIZE, NUM, PN_APP, AP, CPN_H,ADJ_PRE, PP, NUM_PRE
G12	S1, SIZE, PN_APP, NUM, AP, CPN_H,ADJ_PRE, PP, COP
G123	NO_ANT, NUM, S1, SUP_PRE

The second classification considered the classes *New* (including both *Brand-new* and *Anchored-new*) and *Other*, corresponding to *Old*. The results are presented in Table 6. Results were all higher than for *Brand-new*. G123 shows higher precision and a much higher recall than the other groups. The number of resulting attributes was again 4 in G123 (see Table 7).

6 Feature Analysis

Tables 5 and 7, in the previous section, show the features included in the generated decision trees. The larger number of attributes in a tree was 8 and 9, for G1 and G12. When NO_ANT was considered, this number went down to 4. Features APP, REL_CL, PRON_DET, SUP were never included in the resulting trees. The attributes were evaluated separately to verify which of them contributed individually and strongly for the classification.

The prominent features for *Brand-new* DD classification of each group are displayed in Table 8. In G1, SIZE was a feature that, alone, was able to reach 44% F-measure, with 67% precision. S1 in G2, although has shown 100% precision, is of limited recall, since it only applies to the first sentence of each text. In G3, NO_ANT had 73% F-measure and 64% precision. The SIZE feature is an original attribute that is simple to be verified and has presented a significant precision result if compared to the entire group G1 and also with higher precision than NO_ANT of G3. For these reasons, we analyzed decision trees generated on the basis of G1 but without the SIZE feature (G1 without SIZE), in Table 9. We noticed that the feature SIZE replaces other features commonly present in related work (prepositional phrases, relative clauses) in a satisfactory way and presents increases in the number of correctly classified descriptions and in precision in the classification of *Brand-new* DDs. When SIZE is not considered, the resulting tree includes PP, ADJ_PRE, REL_CL, which didn't appear before.

Table 8: Feature analysis
Precision (P); Recall (R); F-measure (F)

Feature Alone	**P**	**R**	**F**
SIZE (G1)	67%	33%	44%
S1 (G2)	100%	6%	11%
NO_ANT (G3)	64%	86%	73%

Table 9: Feature SIZE
Correct C); Precision (P); Recall (R); F-measure (F)

Features	**C**	**P**	**R**	**F**
G1	63%	65%	55%	60%
G1 without SIZE	62%	63%	58%	61%

In the *New* DD classification, the only feature that presented a distinction when applied alone was NO_ANT with 76% of correct classification, precision of 76% and recall of 86%. Other features alone were not able to distinguish the examples. When the previous text is considered as a feature, the features related to the noun phrase structure seem to loose their importance for the task.

7 Evaluation on unseen data

The decision trees learned in the experiments shown in the last section were applied to completely unseen data - Corpus 2. So we could also check the adequacy of the learned trees for this variant of Portuguese. The results are presented below.

The results of the *Brand-new* classifier applied to Corpus 2 can be seen in Table 10. We adopted as baseline (B) an algorithm that classifies all definite descriptions as *Brand-new*. As before, group G123 showed the best results. The difference from G123 to G1 and G12 was significant (99.5%). We verified significant gains in precision (from 60% to 86%) and F-measure (from 75% to 80%) considering the given baseline. Note that for the *Other* class, F-measure was never lower than 66%. G1 alone shows improvements in precision compared to the baseline (from 60% to 80%).

Table 10: Brand-new Classification
Correct(C); Precision (P); Recall (R); F-measure (F)

Classes	**B**				**G1**				**G12**				**G123**			
	C	**P**	**R**	**F**	**C**	**P**	**R**	**F**	**C**	**P**	**R**	**F**	**C**	**P**	**R**	**F**
B-new	59%	60%	100%	75%	68%	80%	62%	70%	69%	80%	64%	71%	78%	86%	76%	80%
Other		0	0	0		58%	77%	66%		59%	76%	66%		70%	82%	75%

For the class *New*, the results of Group G123 are significantly higher than the others (99.5%), 83% of precision and 85% of F-measure, against a baseline of 64%, and 78% (see Table 11). Again, group G1 presents improvements in comparison to the baseline (from 64% to 80%).

The results reported are even better than the ones shown for the learning phase, this is probably related to the higher number of *Brand-new* and *New* DDs in the European Portuguese Corpus (Table 2). Features related to the noun phrase structure have been used in many of the previous work, and we can see here that they alone can indicate, with considerable precision, the novelty level of DDs.

Table 11: New Classification
Correct (C); Precision (P); Recall (R); F-measure (F)

Classes	**B**				**G1**				**G12**				**G123**			
	C	**P**	**R**	**F**	**C**	**P**	**R**	**F**	**C**	**P**	**R**	**F**	**C**	**P**	**R**	**F**
New	64%	64%	100%	78%	65%	80%	61%	70%	67%	79%	66%	72%	81%	83%	88%	85%
Other		0	0	0		52%	73%	61%		54%	70%	61%		76%	69%	72%

8 Final Remarks

This work presented the evaluation of a classification system of *Brand-new* and *New* DDs for Portuguese. The evaluation was carried out on completely unseen data. The results were stable. Classifying *New* DDs seems to be easier than classifying *Brand-new* DDs, as we can see higher F-measure values for this class (although this was clearer in the first experiments with corpus 1). In the classification of *Brand-new* DDs, Group G123 has shown a F-measure of 80%. Group G1 has shown a precision of 80%. Group G12 doesn't show much improvement due to the limited number of cases in copular constructions and in first sentences. In the classification of *New* DDs, the attributes in G123 showed a F-measure of 85%. In G1, the precision is 80%, near to 83% seen in G123.

We were interested in the contribution of the noun phrase alone for the classification (G1), and we found that it was indeed enough for achieving high precision. These findings might have interesting consequences for other tasks, such as summarization. In an extracted summary, for instance, DDs can be analyzed solely according to their intrinsic structure, to verify if they are new in the discourse. In these cases they would not bring problems of coherence to the summary due to the lack of an antecedent.

A detailed evaluation of the features was made. We found that the feature SIZE alone presented a better precision than other features in Group 1 altogether (67%). This feature seems to replace well several complex syntactic features often used in other systems, such as relative clauses and prepositional phrases. It is a simple feature that has not been mentioned in previous work so far. The feature NO_ANT (G3) was rather relevant in both classifications, confirming the findings of [7] for English. In fact, when classifying *New* DDs it is the only salient feature. Also, this feature minimizes the importance of other features. Indeed, looking for the presence of an identical antecedent seems to do alone most of the job.

We acknowledge that related work deal with different kinds of NPs, different features, languages and data. This of course makes the comparison difficult. However, we can see that, in general, the results of the proposed system are not far from the state of the art in the area as reported by previous work (Table 12). From a initial set of 16 features our classifier achieved best measures on the basis of 4 of them. As future work, we intend to carry out an investigation into other romance languages and other classes (*Plain-old, Related-old, Anchored-new*).

Table 12: Related work

Related Work	**P**	**R**	**F**	**#Features**
[2] - Independent existential DDs	87%	78%	82%	10
[14] - Discourse new DDs	72%	69%	70%	9
[13] - Discourse new DDs	85%	82%	83%	32
[8] - Discourse new DDs	95%	86%	90%	9
We - Brand-new DDs	86%	76%	80%	16/4
We - New DDs	83%	88%	85%	16/4

References

1. C. Aone and S. Bennett. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 122–129, Cambridge, Massachusetts, USA, 1995.
2. D. L. Bean and E. Riloff. Corpus-based indentification of non-anaphoric noun phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 373–380, College Park, Maryland, USA, 1999.
3. E. Bick. *The Parsing System PALAVRAS: Automatic Grammatical Analysis of Protuguese in a Constraint Grammar Framework*. PhD thesis, Arhus University, Arhus, 2000.
4. R. M. Guillena, M. Palomar, and A. Ferrández. Processing of spanish definite descriptions. In *Proceedings of the Mexican International Conference on Artificial Intelligence*, pages 526–537. Springer-Verlag, 2000.
5. C. Müller, S. Rapp, and M. Strube. Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 352–359, Philadelphia, PA, 2002.
6. C. Müller and M. Strube. Mmax: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 45–50, Seattle, Washington, 2001.
7. V. Ng. Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 151–158, Barcelona, Spain, 2004.
8. M. Poesio, M. Alexandrov-Ksbadjov, R. Vieira, R. Goulart, and O. Uryupina. Does discourse-new detection help definite description resolution? In *Proceedings of the 6th International Workshop on Computational Semantics*, pages 236–246, Tiburg, 2005.
9. M. Poesio and R. Vieira. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216, 1998.
10. E. F. Prince. Toaward taxonomy of given-new information. In *P. Cole, editor Radical Gramatics*, pages 223–256, New York, 1981. Academic Press.
11. S. Salmon-Alt and R. Viera. Nominal expressions in multilingual corpora: Definites and demonstratives. In *Proceedings of the LREC*, pages 1627–1634, Las Palmas, 2002.
12. W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. In *Computational Linguistics*, volume 27, pages 521–544, 2001.
13. O. Uryupina. High-precision identification of discourse new and unique noun phrases. In *Proceedings of the 41st Annual Meeting on ACL*, pages 80–86, Sapporo, Japan, 2003.
14. R. Vieira and M. Poesio. An empirically-based system for processing definite descriptions. *Computational Linguistics*, 26(4):525–579, 2000.

Analysing Definition Questions by Two Machine Learning Approaches

Carmen Martínez and A. López López
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro # 1
Santa María Tonanzintla, Puebla, 72840, México
carmen@inaoep.mx, allopez@inaoep.mx

Abstract

In automatic question answering, the identification of the correct target term (i.e. the term to define) in a definition question is critical since if the target term is not correctly identified, then all subsequent modules have no chance of providing relevant nuggets. In this paper, we present a method to tag a question sentence experimenting with two learning approaches: QTag and Hidden Markov Model. We tested the methods in five collections of questions, PILOT, TREC 2003, TREC 2004, CLEF 2004 and CLEF 2005. We performed ten-fold cross validation for each collection and we also tested with all questions together. The best accuracy rates for each collection were obtained using QTag, but with all questions together the best accuracy rate is obtained using HMM.

1. Introduction

Question Answering (QA) is a computer-based activity that tries to improve the output generated by Information Retrieval (IR) systems, and involves searching large quantities of text and ”understanding” both questions and textual passages, to the degree necessary to recommend a text fragment as an answer to a question.

Regarding the input of QA systems, according to [1] there are five sorts of questions:

1. Factual questions. The answer is a number, short phrase or sentence fragment obtained from one document (e.g. When was the telegraph invented?).

2. List questions. The answer is a list of an exact number of short phrases or sentence fragments from different documents (e.g. Name 20 countries that produce coffee).

3. Definition questions. The answer is a list of complementary short phrases or sentence fragments from different documents (e.g. What are nanoparticles?, Who was Christopher Reeve?).

4. Complex questions. The question is separated in sub-questions so, to answer the complex question, the sub-questions have to be answered first (e.g. How have thefts impacted on the safety of Russia’s nuclear navy, and has the theft problem been increased or decreased over time? a) What specific instances of theft do we know about? . . . e) What is meant by nuclear navy?).

5. Speculative questions. To answer this kind of question, it is necessary some kind of reasoning (e.g. Is the airline industry in trouble?).

There are seven interrogative adverbs (*who, why, how, which, what, where, when*), from these only *what* and *who* can be interrogative adverbs for definition questions since they express a request for *a concise explanation of the meaning of a word, phrase, symbol or explanation of the nature of a person or thing*.

Who can be used to formulate both factual and definition questions. So, if a question is *who is the president of Mexico?*, this is not a definition question since just requires a name, but *who is Vicente Fox?*, demands an explanation about a specific person.

Usually, when we talk about a definition we mean a sentence or a paragraph. For instance, a definition of *nugget* would be *a solid lump of a precious metal (especially gold) as found in the earth.* But according to the current state of the art in definition question answering [2], the reply is a set of only sentence fragments (precisely called nuggets). So, for the example "nugget", the answer can be the following fragments: *a solid lump, precious metal, gold, earth*.

When evaluating systems answering definition questions, a set of terms are given by assessors, who developed the questions. Also, these topics are given already classified as *vital* (important) and *ok* or non vital (less important).

Nowadays, definition questions have drawn much attention [2]. Answering definition questions is different to answering factual questions, as we described above, since in definition questions, there are several vital and non vital nuggets. In contrast, in factual questions the answer is a unique number, short phrase, or sentence fragment. Two representative works to definition questions answering are: Hildebrandt et. al. [3] presented a multi-strategy approach using a database constructed offline with surface patterns, a Web-based dictionary, and an off-the-shelf document retriever. They employed a simple pattern-based parser using regular expressions to analyze the questions. On the other hand, Tsur [4] used text categorization and a biography learner to improve the task, i.e. definition question answering. Questions analysis is rather naive based on keywords, articles, determiners, capitalization, and name recognition.

For all definition question systems, the first module is target extraction, i.e. the term to define. However, some authors [5, 6] that present an analysis of their errors, found that they obtained poor efficacy because many errors can be traced back to problems with target extraction. If the target term is not correctly identified, then all subsequent modules have no chance of providing relevant nuggets. So, given the question, a key problem to resolve is to obtain the target term since this will be the term to define. For instance, in the following questions:

What are nanoparticles?
Who is Niels Bohr?
What is Friends of the Earth?
Who was Abraham in the Old Testament?

Nanoparticles, Niels Bohr, Friends of the Earth and Abraham are target terms. We can identify three different structures of questions: when the target is a single term, e.g. a noun, when the target is a named entity, and when the target term comes with some other words that are possibly its context.

The main idea to analyze the definition question and obtain the target term and additional information (context terms) is: the interrogative adverb and the verbal form are removed from each question. Then, we apply a named entity tagger, if the result is only one word or one named entity, then there is no choice, that is the target term. For the rest of the questions, we apply a Part-Of-Speech (POS) tagger. From this, the idea is to check if the question follows a previously found pattern that can immediately reveal the target and context terms. To achieve this, we have to tag previously the known sentences to obtain a training set and make a special purpose tagger, i.e. a *question sentence tagger*. The principal tags that we used are *V*, for terms that can be ignored, *T* for the target term, and *C* for context terms.

The paper is organized as follows: next section describes briefly the learning algorithms: Hidden Markov Model (HMM) and QTag; Section 3 presents the method to tag question sentences; Section 4 reports experimental results; finally, some conclusions and directions for future work are presented in Section 5.

2. Learning Algorithms

In this section, we describe the two Machine Learning approaches, Hidden Markov Model and QTag, that we applied to solve the problem.

2.1. Hidden Markov Model

A Hidden Markov Model (HMM), as Rabiner describes in [7], is a Markov chain, where each state generates an observation. An HMM is specified by a five-tuple (*S*,*K*,Π,*A*,*B*), where *S* is the set of states, *K* the output alphabet and Π, *A*, *B* are the probabilities for the initial state, state transitions, and symbol emission, respectively.

Given appropriate values of *S*,*K*,*A*,*B*, and Π, the HMM can be used as a generator to return an observation sequence $O = O_1 O_2 \cdots O_T$
where each observation O_t is one of the symbols from *B*, and *T* is the number of observations in the sequence.

There are three basic questions that we want to know about an HMM:

1. Given the observation sequence $O = O_1 O_2 \cdots O_T$ and a model $\lambda = (A, B, \Pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?.
2. Given the observation sequences *O*, and the model λ, how do we choose a corresponding state sequence $Q = q_1 q_2 \cdots q_t$ which is optimal in some meaningful sense (i.e., best "explains" the observations)?
3. How do we adjust the model parameters λ to maximize $P(O|\lambda)$?

In question 1, given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model. Question 2 is intended to uncover the hidden part of the model, i.e., to find the "correct" state sequence. Question 3 points to the process to optimize the model parameters to best describe how a given observation sequence is generated.

2.2. Applying HMMs to POS tagging

HMMs can be used to POS tagging but for this task, parameters can not be randomly initialized, since this would leave the tagging task too unconstrained. The symbol emission probabilities is initialized using the method of Jelinek [8]:

$$b_{j.l} = \frac{b^{\star}_{j.l} C(w^l)}{\sum_{w^m} b^{\star}_{j.m} C(w^m)}$$

where the sum is over all words w^m in the dictionary, and

$$b^{\star}_{j.l} = \begin{cases} 0 & \text{if } t^j \text{ is not a part of speech allowed for } w^l \\ \frac{1}{T(w^l)} & \text{otherwise} \end{cases}$$

where $T(w^j)$ is the number of tags allowed for w^j.

2.3. QTag

QTag [9] is a robust probabilistic parts-of-speech tagger. This is a program that reads text and, for each token in the text, returns the part-of-speech (e.g. noun, verb, punctuation, etc). QTag was advantageous for our needs because we can create our own resource files for a different language or tagset, we simply supply a pre-tagged training corpus. The size of the training data is obviously important for the accuracy of the tagging procedure.

3. The Method to Tag Question Sentences

The process to obtain the target term is the following:
We remove the interrogative adverb (*who* or *what*) and the verbal form (*is*, *are* or *was*) from each question. For example, from the questions given above, we get:

nanoparticles?
Niels Bohr?
Friends of the Earth?
Abraham in the Old Testament?

Then, we apply a named entity tagger (LingPipe) [10]. For the same questions, we obtain the following:

nanoparticles?
< type="PERSON" Niels Bohr > ?
Friends of the < type="LOCATION" Earth > ?
Abraham in the Old < type="PERSON" Testament > ?

CC	Coordinating Conjunction
CD	Cardinal number
DT	Determiner
IN	Preposition or Subordinating conjunction
JJ	Adjective
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
’	’
,	,
”	”
”	”
(	(
)	)
?	end

Table 1. Subset of tags produced by MBT.

V	void
T	target
C	context
’	’
,	,
”	”
”	”
(	(
)	)
?	end

Table 2. Tags used by the Question Sentence Tagger.

If the result is a single word or a named entity (as the first and second examples), then that is the target term. For the rest of the questions, we apply a Part-Of-Speech (POS) tagger, in our case Memory Based Tagging (MBT) [11] that, by the way, has a better performance tagging English questions than QTag. Table 1 details the subset of tags obtained so far. For the examples that we are using to illustrate and two additional examples, we obtain:

Friends/NNPS of/IN the/DT Earth/NNP ?/.
Abraham/NNP in/IN the/DT Old/NNP Testament/NNP ?/.
Treasury/NNP Secretary/NNP
Robert/NNP Rubin/NNP ?/.
the/DT International/NNP Committee/NNP
of/IN the/DT Red/NNP Cross/NNP ?/.

By tagging the sentence with part-of-speech, we generalize and work thereafter with patterns of questions, rather than raw text. Named entities within a context are also processed in this way (as noun phrases) in order to identify simultaneously target (named entity) and context. For the examples, we keep the following sequences of tags:

NNPS IN DT NNP ?
NNP IN DT NNP NNP ?
NNP NNP NNP NNP ?
DT NNP NNP IN DT NNP NNP ?

Then we tagged these sequences of part-of-speech labels according to our needs to obtain a training set and reach a special purpose tagger, i.e. a *question sentence tagger*. This is done in two ways, using QTag and HMM. Table 2 shows tags used by the *question sentence tagger*.

For the previous examples, we have

NNPS/T IN/V DT/V NNP/C ?/?
NNP/T IN/V DT/V NNP/C NNP/C ?/?
NNP/C NNP/C NNP/T NNP/T ?/?
DT/V NNP/T NNP/T IN/V
DT/V NNP/C NNP/C ?/?

These examples are part of the training set. Now if we have a new question, Who is Akbar the Great?, we apply the previous process:

Who is Akbar the Great?
Akbar the Great?
Akbar the < type="ORGANIZATION" Great > ?
Akbar/NNP the/DT Great/NNP ?/.
NNP DT NNP ?

The last sequence (NNP DT NNP ?) is tagged by the *question sentence tagger* using QTag or HMM. The correct tags are: NNP/T DT/V NNP/C ?/?, since "Great" serves as a context helpful to focus the search for a definition of the target term "Akbar".

4. Experimental Setting

We used definition questions from five collections:

COLLECTION	Simple	Complex	Total
PILOT	17	8	25
TREC 2003	31	19	50
TREC 2004	36	29	65
CLEF 2004	65	25	90
CLEF 2005	26	24	50
Total	175	105	280

In this table, by "simple" we refer to the questions where the target term is a single word or named entity and "complex" when the target term comes with some other words that are possibly its context. The collection PILOT [2] contains questions used in the pilot evaluation of definition questions performed by NIST and AQUAINT program contractors. TREC 2003 and TREC 2004 are sets of definition questions used to evaluate Questions Answering systems in the Text REtrieval Conference [12] in 2003 and 2004 respectively. The collections CLEF 2004 and CLEF 2005 are questions obtained from the Cross Language Evaluation Forum [13] in 2004 and 2005.

As mentioned above, we developed also a *question sentence tagger* using Hidden Markov Models (HMM) and QTag. An HMM is specified by a five-tuple (*S*, *K*, Π, *A*, *B*), where S is the set of states (each state is a tag), K the output alphabet, Π the initial state probabilities, A the state transition probabilities and B the symbol emission probabilities. The values used when the collection is ALL, are:

S= { BEGIN, V, T, C, ', ', ", ", (,), ? }
K= { BEGIN, DT, NNP, NN, NNPS, NNS, JJ,
IN, CC, CD,', ', ",",(,), ? }
Π = { 1 0 0 0 0 0 0 0 0 0 0 0 }

The transition probabilities (*A*) are generated randomly and improved with the training examples and *B* is initialized using the method of Jelinek described in the section 2.2. In order to form the training set used by QTag, we tagged the sequences of part-of-speech labels with the tags shown in Table 2, those tags are the same of the set *S* in the HMM.

	Complex	Global
PILOT	62.50	88
TREC 2003	33.33	76
TREC 2004	31.03	69.23
CLEF 2004	84	95.56
CLEF 2005	91.66	96
Average	60.50	84.96

Table 3. Comparison of the accuracy rates of the Question Sentence Tagger using QTag

	Complex	Global
PILOT	50	84
TREC 2003	27.78	74
TREC 2004	24.14	66.15
CLEF 2004	80	94.44
CLEF 2005	83.33	92
Average	53.05	82.12

Table 4. Comparison of the accuracy rates of the Question Sentence Tagger using HMM

5. Results

We performed two different experiments. In the first experiment, we tested separately each collection of questions, Table 3 shows the accuracy rates using QTag and the Table 4 shows the accuracy rates using HMM. In all tests, we made a ten-fold cross validation and the results are the average of five runs.

From the first experiment, we can observe that QTag performs better than HMM on the questions of interest, possibly because that is the kind of processing it was designed for. On the other hand, HMM performs poorly, caused by the small size of the training sets.

In the second experiment, we joined four collections of questions, PILOT, TREC 2003, TREC 2004, CLEF 2004 to form the collection that we called ALL. The collection ALL_1 contains the questions from the five collections. The collection ALL can be used as baseline since we can test if our method improves its performance when the training set increases. Table 5 shows the accuracy rates using QTag and the Table 6 displays the accuracy rates using HMM. Also we performed a ten-fold cross validation for each test.

The results of the second experiments show that HMM behaves better than QTag, from the beginning, with an increased training set. However, QTag is more sensitive to the increment in size of the training set, reflected in a higher percentage of improvement.

As one can observe, the results show that the method is feasible and delivers an acceptable level of accuracy for both approaches. As we increase the training set of question patterns, we expect to increase also the accuracy identifying target and context terms.

Our questions sentence tagger, in either version, had trouble tagging sentences with patterns under-represented. From very few examples, the pattern can not be learnt properly during training. Two instances of this kind of patterns are:

what is Micro Compact Car (MCC)?
NNP NN NN ?
what is the Order of the Solar Temple?
DT NNP IN DT NNP NNP ?

This problem will be overcome as the size of the training set increases.

	Complex	Global
ALL	38.75	78.70
ALL_1	51.43	81.80
% of Improvement	32.72	3.94

Table 5. Comparison of the accuracy rates of the Question Sentence Tagger using QTag

	Complex	Global
ALL	51.25	83.04
ALL_1	60	85
% of Improvement	17.07	2.36

Table 6. Comparison of the accuracy rates of the Question Sentence Tagger using HMM

6. Conclusions and Future Work

We have presented a method to identify the target term in an automatic, fast and flexible way. The method can be extended easily for new complex questions. As far as we know, definition question analysis has not been approached as a special tagging task, and given the results, seems very promising since questions are usually short and following certain patterns.

Moreover, with this method, we have additional information for the search of passages or documents for the answer, since the method identifies the target term along some other terms that are the context and valuable to refine the search for the definition.

Another advantage of our approach with a special purpose tagger is that we do not depend completely on a named entity tagger, specially in complex questions. For instance, the tagger can miss a named entity within a context, but the question tagger can identify target and context adequately.

Future work includes extending the corpus to train, and explore ensemble methods to improve the special purpose tagging. And finally, we have to integrate this method to the complete process of definition questions answering.

7. Acknowledgements

This work was partially supported by a CONACyT research grant U39957 and the scholarship 157233 for the first author.

References

[1] Dan Modolvan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu. Performance Issues and Error Analysis in an Open-Domain Question Answering System. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 33–40, Philadelphia, July 2002.

[2] Ellen M. Voorhees. Evaluating Answers to Definition Questions. *NIST*, pages 1–3, 2003.

[3] Wesley Hildebrandt, Boris Katz, and Jimmy Lin. Answering Definition Question Using Multiple Knowledge Sources. In *Proceedings of HLT/NAACL*, pages 49–56, Boston, 2004.

[4] Oren Tsur. Definitional Question-Anwering Using Trainable Text Classifiers. Master's thesis, Institute of Logic Language and Computation, University of Amsterdam, 2003.

[5] S. Harabagiu and F. Lacatusu. Strategies for Advanced Question Answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL*, pages 1–9, 2004.

[6] Jinxi Xu, Ana Licuanan, and Ralph Weischedel. TREC 2003 QA at BBN: Answering Definitional Questions. In *The Twelfth Text Retrieval Conference (TREC 2003)*, pages 28–35, 2003.

[7] Lawrence Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proc. IEEE*, volume 77, pages 257–286, 1989.

[8] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Languaje Processing*. The MIT Press Cambridge, Massachusetts, London, England, 1999.

[9] Oliver Manson. Qtag-A portable probabilistic tagger. *Corpus Research, The University of Birmingham, U.K.*, 1997.

[10] http://www.alias-i.com/lingpipe/index.html.

[11] Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. MBT: A Memory-Based Part of Speech Tagger-Generator. In *Proceedings of the Fourth Workshop on Very Large Corpora, Copenhagen, Denmark,*, pages 14–27, 1996.

[12] http://trec.nist.gov/.

[13] http://www.clef-campaign.org/.

Fuzzy Rule-Based Hand Gesture Recognition*

Benjamín C. Bedregal[1], Antônio C. R. Costa[2] and Graçaliz P. Dimuro[2]

[1] Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte, Brazil
[2] Informatics School, Graduate Programme in Computer Science
Catholic University of Pelotas, Brazil
bedregal@dimap.ufrn.br, {rocha,liz}@ucpel.tche.br

Abstract. This paper introduces a fuzzy rule-based method for the recognition of hand gestures acquired from a data glove, with an application to the recognition of some sample hand gestures of LIBRAS, the Brazilian Sign Language. The method uses the set of angles of finger joints for the classification of hand configurations, and classifications of segments of hand gestures for recognizing gestures. The segmentation of gestures is based on the concept of monotonic gesture segment, sequences of hand configurations in which the variations of the angles of the finger joints have the same sign (non-increasing or non-decreasing). Each gesture is characterized by its list of monotonic segments. The set of all lists of segments of a given set of gestures determine a set of finite automata, which are able to recognize every such gesture.

1 Introduction

Fuzzy set theory [1] is the oldest and most widely reported component of present-day soft computing (or computational intelligence), which deals with the design of flexible information processing systems [2], with applications in control systems [3], decision making [4], expert systems [5] etc. The significance of fuzzy set theory in the realm of pattern recognition was justified in [2].

A fuzzy system encompasses the implementation of a (usually nonlinear) function, defined by a linguistic description of the relationship between its input variables. Standard fuzzy systems presents an architecture such as the one depitecd in Fig. 1. The *fuzzificator* is the component that computes the membership degrees of the crisp input values to the linguistic terms (fuzzy sets) associated to each input linguistic variable. The *rule base* is composed by inference rules associating linguistic terms of input linguistic variables to linguistic terms of output linguistic values. The *information manager* is the component for searching in the rule base the adequate rules to be applied for the current input. The *inference machine* gives the membership degrees of the output values in the output sets, by the application of the rules selected in the rule base.

* This work is partially supported by FAPERGS and CNPq (Proc. 470871/2004-0, Proc. 470556/2004-8).

Finally, the *defuzzificator* determines a single output value as a function of the output values and their membership degrees to the output sets.

We remark, however, that there are many approximate methods (e.g., classification or pattern recognition procedures) that do not produce a single final result. On the contrary, they may give several alternative solutions to a single problem (e.g., the different classes to which a given input may belong). Examples of such methods are several fuzzy methods for pattern recognition [2], such as fuzzy relations, fuzzy clustering, fuzzy neural systems etc. [6], with applications to signature verification [7], and face recognition [8], for example. Thus, for some specific applications, it is reasonable to consider a fuzzy rule based method, which determines a system architecture as shown in Fig. 2.

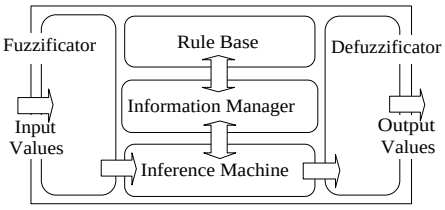

Fig. 1. Architecture of standard fuzzy systems.

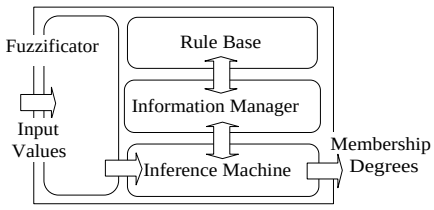

Fig. 2. Architecture of a fuzzy rule based system.

It is possible to find an extensive literature about methods and systems for gesture recognition in general, and hand gesture recognition in particular. There are systems for the recognition of 3-D and 2-D gestures captured by different devices (data gloves, cameras etc.) [9], systems for the graphical recognition of traces left on tablet devices [10] etc. Among several methods for gesture recognition, there are methods based on fuzzy logic and fuzzy sets, methods based on neural networks, hybrid neuro-fuzzy methods [11], fuzzy rule [12] and finite state machine [13] based methods, methods based on hidden Markov models [14] etc. In particular, considering methods for sign language recognition, some literature can be found related to fuzzy methods, such as, for example, fuzzy decision trees [15] and neuro-fuzzy systems [16].

In this paper, we propose a fuzzy rule-based method for the recognition of hand configurations and hand gestures acquired from a data glove, with an application to the recognition of some sample hand gestures of LIBRAS, the Brazilian Sign Language [17]. The method uses the set of angles of finger joints for the classification of hand configurations, and classifications of sequences of hand configurations for recognizing gestures. The segmentation of gestures is based on the concept of *monotonic gesture segment*, sequences of gestures in which the variations of the angles of the finger joints have the same sign (non-increasing or non-decreasing).

Any monotonic gesture segment is characterized by an initial hand configuration, a terminal hand configuration and a list of relevant intermediate

configurations. Each gesture is characterized by a list of monotonic segments. That set of lists of segments determine a set of finite automata, which are able to recognize the gestures being considered.

The paper is organized as follows. In Sect. 2, we introduce our fuzzy rule-based method for hand gesture recognition. A case study is discussed in Sect. 3, with the recognition of LIBRAS hand gestures. Section 4 is the Conclusion.

2 The Fuzzy Rule Based for Hand Gesture Recognition

The objective is to recognize some hand gestures with data obtained from a data glove. Consider a hypothetical data glove with 15 sensors, as shown in Fig. 3. The fingers are labelled as: F1 (little finger), F2 (ring finger), F3 (middle finger), F4 (index finger) and F5 (thumb). The joints in the fingers are labelled as J1 (the knuckle), J2 and J3, for each finger. A separation between two fingers is labelled as Sij to indicate that it is a separation between the fingers Fi and Fj.

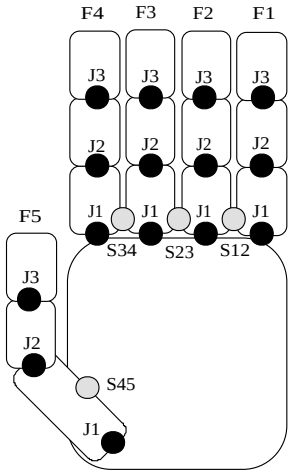

Fig. 3. Localization of sensors in the data glove.

Since any movement can be represented as a sequence of frames, a hand movement using a data glove is represented as a sequence of hand configurations, one for each discrete time instant. That is, at each time instant, the data glove sensors should provide the set of angles of joints and finger separation that characterizes a hand configuration.

In order to simulate this data transfer, a random generator of hand configurations was implemented, generating at each instant one hand configuration represented by a tuple of angles corresponding to each sensor shown in Fig. 3:

```
( (F1J1,F1J2,F1J3), S12, (F2J1,F2J2,F2J3), S23, (F3J1,F3J2,F3J3), S34,
  (F4J1,F4J2,F4J3), S45, (F5J1,F5J2,F5J3) )
```

Given a hand configuration c and a sensor s, denote the value of each sensor angle by $s(c)$, e.g., F1J1(c), S45(c) etc.

2.1 Fuzzification

To each sensor corresponds a linguistic variable, whose values are linguistic terms representing typical angles of joints and separations. For the joints in the fingers (linguistic variables F1J1, F1J2, F1J3 etc.) the linguistic terms are: STRAIGHT, CURVED and BENT. For the separations between fingers F1 and F2, F2 and F3, F4 and F5 (linguistic variable S12, S23, S45), the linguistic terms are: CLOSED, SEMI-OPEN and OPEN. For the separations between fingers F3 and F4 (linguistic variable S34), the linguistic terms are: CROSSED, CLOSED, SEMI-OPEN and OPEN. Tables 1 and 2 present the notations used for linguistic terms of linguistic variables representing joints and separations, respectively. Figures 4, 5, 6, 7 and 8 show the fuzzification adopted for those variables.

Table 1. Linguistic terms of linguistic variables representing finger joints.

Linguistic Term	Notation
STRAIGHT	St
CURVED	Cv
BENT	Bt

Table 2. Linguistic terms of linguistic variables representing finger separations.

Linguistic Term	Notation
CROSSED	Cr
CLOSED	Cd
SEMI-OPEN	SOp
OPEN	Op

The hand configuration is the main linguistic variable of the system, denoted by HC, whose linguistic terms are names of hand configurations, which names are application dependent. For instance, in Sect. 3, names of Brazilian Sign Language (LIBRAS) hand configurations (see Fig. 10) were used for such linguistic terms.

2.2 The Recognition Process

The hand gesture recognition process is divided into four steps: (1) recognition of finger configurations, (2) recognition of hand configurations, (3) segmentation of the gesture in monotonic hand segments and (4) recognition of the sequence of monotonic hand segments.

For the step 1 (*recognition of finger configurations*), 27 possible finger configurations are considered. These configurations are codified in the following format: XYZ, where X is the value of the linguistic variable corresponding to the first joint J1, Y is the value of the linguistic variable corresponding to the second joint J2 and Z is the value of the linguistic variable corresponding to the third joint J3. For example, StStSt is used to indicate that the three joints are

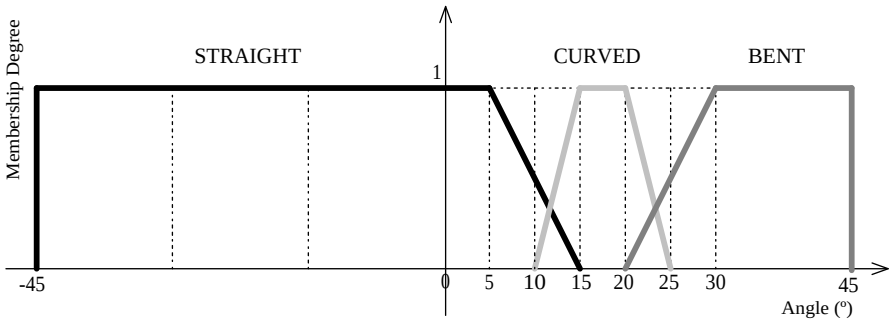

Fig. 4. Fuzzification of the linguistic variable of the joint F5J2 in the thumb finger F5.

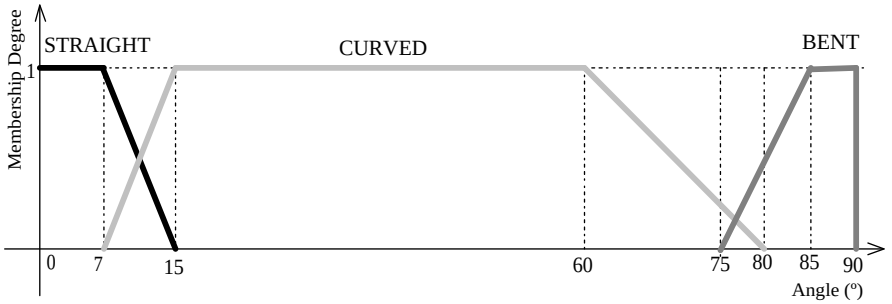

Fig. 5. Fuzzification of the linguistic variables of remaining finger joints.

STRAIGHT, StCdCd indicates that the first joint is STRAIGHT whereas the others are CURVED etc.

The 27 possible finger configurations determine 27 inference rules that calculate membership degree of each finger to each configuration. For example:

```
If F4J1 is STRAIGHT and F4J2 is CURVED and F4J3 is CURVED
Then F4 is StCdCd
```

The next step is 2 (*recognition of hand configurations*), where the hand configuration is determined, considering each finger configuration and each separation between fingers. For example, the rule for the hand configuration [G] of LIBRAS (see Fig. 10) is described below:

```
If F1 is BtBtSt and S12 is Cd and F2 is BtBtSt and S23 is Cd and
   F3 is BtBtSt and S34 is Cd and F4 is StStSt and S45 is Cd and
   F5 is StStSt
Then HC is [G]
```

In 3 (*segmentation of the gesture in monotonic hand segments*), we divide each gesture in a sequence of k limit hand configurations $l_1, \ldots, l_k$, where l_1 is the initial gesture configuration and l_k is the terminal gesture configuration.

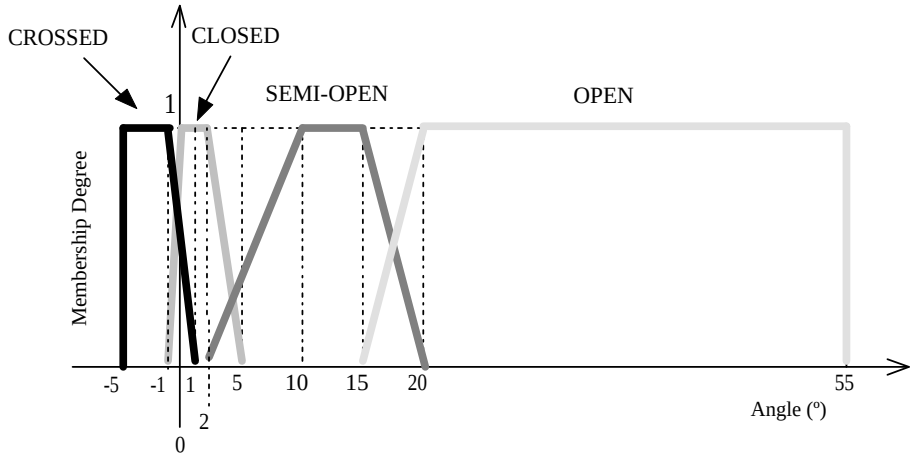

Fig. 6. Fuzzification of the linguistic variable of the separation S34 between the middle finger F3 and the index finger F4.

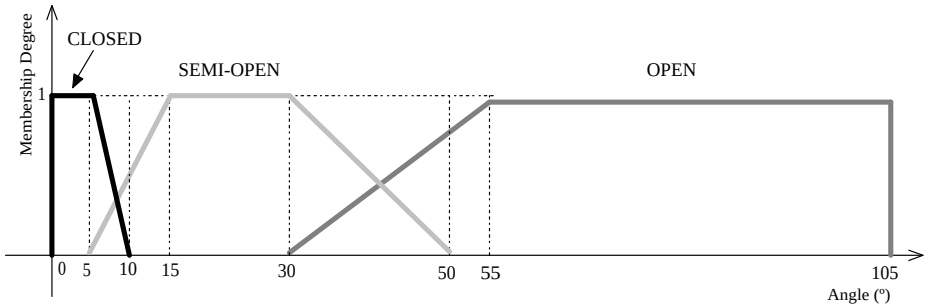

Fig. 7. Fuzzification of the linguistic variable of the separation S45 between the index finger F4 and the thumb finger F5.

The limit configurations are such that, for each c between l_i and l_{i+1}, and for each sensor s, $s(c) - s(l_i)$ has the same sign of $s(l_{i+1}) - s(l_i)$, for $i = 1, \ldots, k-1$ (a difference equal to 0 is compatible with both negative and positive signs).

The limit hand configurations are the points that divide the gesture into monotonic segments, that is, segments in which each sensor produces angle variations with constant (or null) sign. For each monotonic segment $l_i l_{i+1}$, l_i and l_{i+1} are its initial and terminal hand configurations, respectively.

The procedure for step 3 is the following. To find any monotonic segment $l_i l_{i+1}$, the next n configurations sent by the data glove after l_i are discarded, until a configuration c_{n+1}, such that the signs of $s(c_{n+1}) - s(c_n)$ and $s(c_n) - s(l_1)$ are not the same (or, c_{n+1} is the last configuration of the gesture). Then, c_n (resp., c_{n+1}) is the terminal hand configuration l_{i+1} of the considered monotonic segment, and also coincides with the initial configuration of the next segment

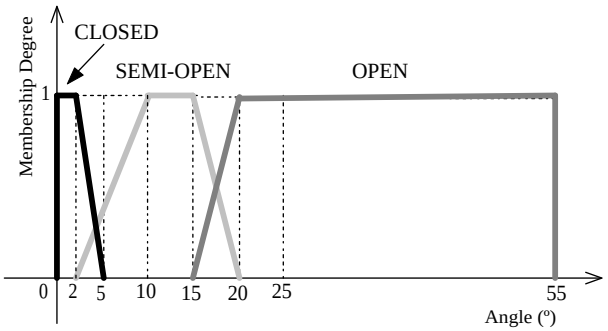

Fig. 8. Fuzzification of the linguistic variables of the separations between remaining fingers.

$l_{i+1}l_{i+2}$ (if there is one). The process starts with $l_i = l_1$, which is the initial gesture configuration, and is repeated until the end of the gesture, generating the list of k limit hand configurations.

In 4 (*recognition of the sequence of monotonic hand segments*), the recognition of each monotonic segment $l_i l_{i+1}$ is performed using a list of reference hand configurations $r_1, r_2, \ldots, r_m$ that characterizes the segment, where r_1 and r_m are the initial and terminal hand configurations of the segment, respectively. A monotonic segment is recognized by checking that it contains its list of reference hand configurations. The process is equivalent to a recognition based on a linear finite automaton (shown in Fig. 9), where $l_i = r_1$ and $l_{i+1} = r_m$.

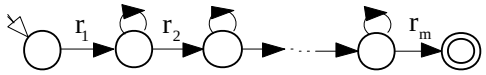

Fig. 9. Automaton for the recognition of monotonic segments.

3 Case Study: Hand Gestures of LIBRAS

As any other sign language (e.g., ASL – American Sign Language, used in the USA), LIBRAS (Língua Brasileira de Sinais – Brazilian Sign Language) is a natural language endowed with all the complexity normally found in the oral-auditive languages. Thus, it can be analyzed at all the various linguistic levels encountered in such languages, such as the "phonetic-phonological" level (also called "cheremic" level, for its relationship with the movement of the hands), the syntactic level, and the semantic and pragmatic levels [17].

As a language of the specific modality called visual-gestural, however, the elements that constitute many of those linguistic levels are of a specific nature.

For instance, the main parameters that characterize the "phonological" units of sign languages are: the configurations of the hands used in the gestures, the main spatial location (relative to the persons who is signing) where the movements of the gestures are performed, the different movements (of the fingers in the hand, of the hands and arms in the space, of the whole body) that constitute the gesture, the facial expressions that express different syntactic, semantic and pragmatic marks during the production of the signs etc.

In the various works on automatic recognition of sign languages that have been developed along the years (see Sect. 1) the recognition of hand gestures has occupied a prominent place. Using capture devices like data gloves and cameras, hand gestures have been analyzed and recognized in order to allow the computer understanding of such basic component of sign languages.

To support that recognition process, a reference set of hand configurations is usually adopted, driven either from the linguistic literature on sign languages, or dynamically developed by the experimenters with an ad hoc purpose. For our purposes, we have chosen a standard set of hand configurations (some of them shown in Fig. 10), taken from the linguistic literature on LIBRAS [17].

Since we take the set of hand configurations from the literature, our method requires that each sign be thoroughly characterized in terms of its monotonic segments and the sequences of hand configurations that constitute such segments, and that the identification of the monotonic segments and hand configurations be manually provided to the system. Of course, a capture device such as a data glove can be used to help to identify the typical values of the angles of the finger joints, but the final decision about the form of the membership functions that characterize the linguistic terms used in the system has to be explicitly taken and manually transferred to the system.

We illustrate here the application of the method by the definition of the necessary parameters for the recognition of the hand gestures that constitute the sign CURIOUS, in LIBRAS. CURIOUS is a sign performed with a single hand placed right in front of the dominant eye of the signer, with the palm up and hand pointing forward. The initial hand configuration is the one named [G1] in Fig. 10. The gesture consists of the monotonic movement necessary to perform the transition from [G1] to [X] and back to [G1] again, such movements been repeated a few times (usually two or three). Thus, a possible analysis of the hand gestures that constitute the sign CURIOUS in LIBRAS is:

Initial configuration: [G1]
Monotonic segment S1: [G1]-[G1X]-[X]
Monotonic segment S2: [X]-[G1X]-[G1]
State transition function for the recognition automaton: see Fig. 11.

To support the recognition of the monotonic segments of CURIOUS, we have chosen to use one single intermediate hand configuration, [G1X]. It is an intermediate configuration that does not belong to the reference set (Fig. 10) and whose characterization in terms of the set of membership functions for linguistic terms was defined in an ad hoc fashion, for the purpose of the recognition of CURIOUS. Together with [G1] and [X], it should be added to the list of hand configurations used by the recognition system.

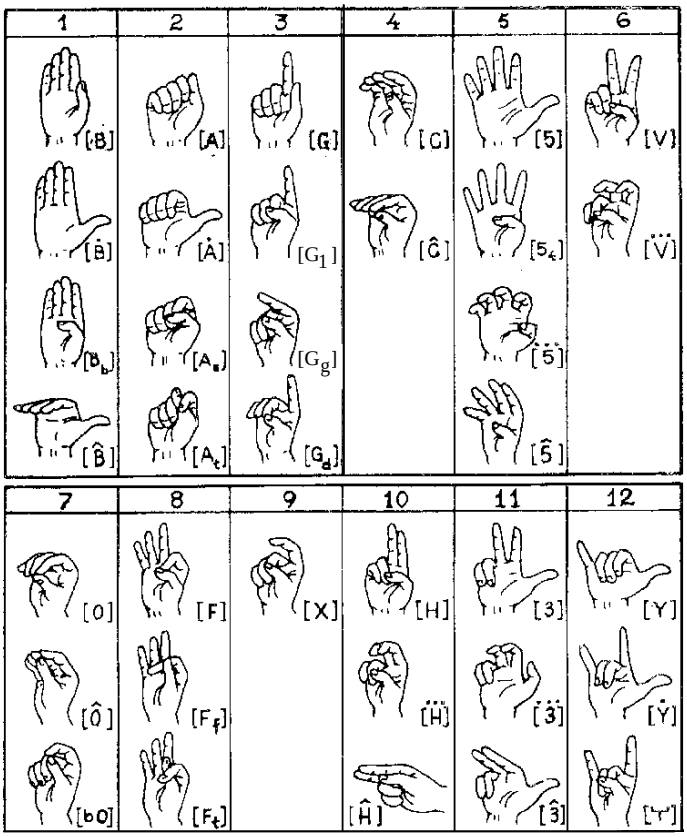

Fig. 10. Some LIBRAS hand configurations.

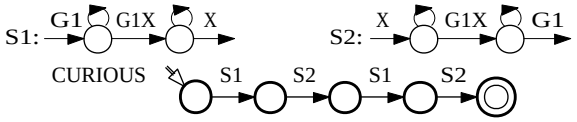

Fig. 11. Automaton for the recognition of hand gestures of the sign CURIOUS.

4 Conclusion and Final Remarks

We presented a fuzzy rule-based for the recognition of hand gestures. The method is highly dependent on a detailed previous analysis of the features of the gestures to be recognized, and on the manual transfer of the results of that analysis to the recognition system. This makes it suitable for the application to the recognition of hand gestures of sign languages, because of the extensive analysis that linguists that have already done of those languages. Prototypes of a random gesture generator and of the gesture recognizer were implemented in

the programming language Python. In the fuzzification process, we considered only trapezoidal fuzzy sets and the minimum (or Gödel) t-norm, motivated by simplicity. Initial experimentation indicated promising results. Future work is concerned with the recognition of arm gestures, by including the analysis of the angles of arm joints.

References

1. L.A. Zadeh, Fuzzt Sets, *Information Control* **8**, 338–353 (1965).
2. S. Mitra, S.K. Pal, Fuzzy Sets in Pattern Recognition and Machine Intelligence, *Fuzzy Sets and Systems* **156**, 381–386 (2005).
3. G. Chen, T.T. Pham, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems* (CRC Press, Boca Raton, 2001).
4. C. Carlsson, R. Fuller, *Fuzzy Reasoning in Decision Making and Optimization* (Physiva-Verlag Springer, Heidelberg, 2002).
5. W. Siler, J.J. Buckley, *Fuzzy Expert Systems and Fuzzy Reasoning* (John Wiley & Sons, Inc., New York, 2004).
6. C.T. Lin, C.S.G. Lee, *Neural Fuzzy Systems: A Neuro-fuzzy Synergism to Intelligent Systems* (Prentice Hall, Upper Saddle River, 1996).
7. M. Hanmandlu, M.H.M. Yusof, V.K. Madasu, Off-line Signature Verification and Forgery Detection Using Fuzzy Modeling, *Pattern Recognition* **38**(3), 341–356 (2005).
8. K. Kwak, W. Pedrycz, Face Recognition Using a Fuzzy Fisherface Classifier, *Pattern Recognition* **38**(10), 1717–1732 (2005).
9. O. Bimber, Continuous 6DOF Gesture Recognition: a Fuzzy-logic Approach, in: Proc. of VII Intl. Conf. in Central Europe on Computer Graphics, Visualization and Interactive Digital Media, V. 1. (1999), pp. 24–30
10. J. Ou, X. Chen, J. Yang, Gesture Recognition for Remote Collaborative Physical Tasks Using Tablet PCs, in: Proc. of IX IEEE Intl. Conf. on Computer Vision, Work. on Multimedia Technologies in E-Learning and Collaboration (Nice, 2003).
11. N.D. Binh, T. Ejima, Hand Gesture Recognition Using Fuzzy Neural Network, in: Proc. ICGST Conf. Graphics, Vision and Image Proces (Cairo, 2005), pp. 1–6.
12. M. Su, A Fuzzy Rule-based Approach to Spatio-temporal Hand Gesture Recognition, *IEEE Transactions on Systems, Man and Cybernetics, Part C* **30**(2). 276–281 (2000).
13. P. Hong, M. Turk, T.S. Huang, Gesture Modeling and Recognition Using FSM, in: Proc. of IEEE Conf. Face and Gesture Recognition (Grenoble, 2000), pp. 410–415.
14. G. Rigoll, A. Kosmala, S. Eickeler, High Perfomance Real-time Gesture Recognition Using Hidden Markov Models, in: Gesture and Sign Language in Human-Computer Interaction, Proc. of International Gesture Workshop, Bielefeld, 1997, edited by I. Wachsmuth, M. Frölich (n. 1371 in LNAI, Springer, 1998), pp. 69–80.
15. G. Fang, W. Gao, D. Zhao, Large Vocabulary Sign Language Recognition Based on Fuzzy Decision Trees, *IEEE Transactions on Systems, Man and Cybernertics* **34**(3), 305–314 (2004).
16. O. Al-Jarrah, A. Halawani, Recognition of Gestures in Arabic Sign Language Using Neuro-fuzzy Systems, *Artificial Intelligence* **133**(1–2), 117–138 (2001).
17. L.F. Brito, *Por uma Gramática de Línguas de Sinais* (Tempo Brasileiro, Rio de Janeiro, 1995) (in Portuguese).

Comparison of distance measures for historical spelling variants

S. Kempken, W. Luther, and T.Pilz

Institute of Computer Science and Interactive Systems
Universität Duisburg-Essen
D-47048 Duisburg, Lotharstr. 65, Germany
{kempken, luther, pilz}@informatik.uni-duisburg.de

Abstract. This paper describes the comparison of selected distance measures in their applicability for supporting retrieval of historical spelling variants (hsv). The interdisciplinary project Rule-based search in text databases with nonstandard orthography develops a fuzzy full-text search engine for historical text documents. This engine should provide easier text access for experts as well as interested amateurs. The FlexMetric framework enhances the distance measure algorithm found to be most efficient according to the results of the evaluation. This measure can be used for multiple applications, including searching, post-ranking, transformation and even reflection about one's own language.

1 Introduction

In recent years, many countries have started retro-digitization projects of precious originals. Events like the disastrous fire in the German Herzogin Anna Amalia Library, a World Heritage Site, in September 2004 show plainly the importance of such preservation, at least of the intellectual contents. Furthermore, these projects make accessible historical texts by building digital libraries that are of interest to scholars of all text-focused disciplines (philologists, historians, linguists, etc.) as well as interested amateurs. Right now, more than one hundred scientific initiatives are involved in the digitization of text collections, electronic editions, rare manuscripts, dictionaries, charters and illustrated books. Most of these initiatives provide digitized facsimiles, some offer additional full text. Hockey [11] provides a survey of important international projects.

The amount of time required to build a digital archive is not to be underestimated. Therefore, many retro-digitization projects focus on the constructional steps of the digitization process, which involve digitizing as well as tagging and aligning the text. Subsequent steps, like manual post processing or elaborate search functions, often need to be put at the bottom of the list. Compact Memory, a project for the digitization of historical Jewish periodicals, for example, combines a comely interface with a respectable archive and is well used. But, as it is a publicly funded project, the operator cannot devote his resources to manually revising optical character recognition (OCR) errors in the digitized texts

or to offering advanced search capabilities. A reliable search engine, however, is the means that makes the data fully accessible.

Particularly historical but also regional texts often involve another important problem, apart from OCR errors: they contain spelling variants. German texts prior to 1901, when a major reform of German orthography took place, are not standardized. The result is a reduced recall ratio in those texts, due to queries that do not cover all possible spellings. The frequency of variant spelling increases significantly with the age of the text documents. Figure 1 shows the amount in percent of nonstandard tokens in 35 historical German texts from 1463 to 1876.

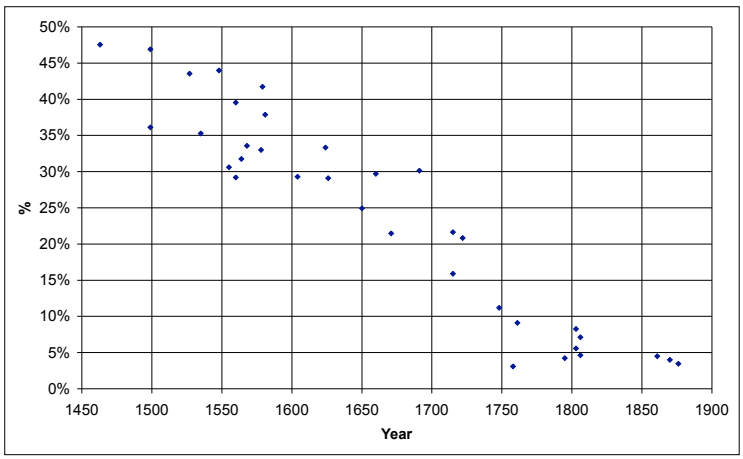

Fig. 1. Frequency of variant spellings in historical text documents

Historical spellings are by no means solely a German problem. Spelling variation is known to occur in English historical corpora also. An initiative by the University Centre for Computer Research on Language (UCREL) of Lancaster University and the University of Central Lancashire (UCLan) has developed a VARiant Detector (VARD) trained on 16th to 19th century data [18].

The interdisciplinary project Rule-based Search in Text Databases with Nonstandard Orthography (RSNSR) supported by the Deutsche Forschungsgemeinschaft (DFG [German Research Foundation]) is currently developing a fuzzy trainable full-text search engine for historical text documents [17]. Since our main focus is the time period from 1700 - 1900, regarding the results shown

in Figure 1, 2 - 25% variant spellings are estimated for those texts. In the worst case, up to one quarter of a text will consist of nonstandard spellings.

In contrast to capacious glossary projects like the *Deutsches Rechtswörterbuch (DRW)* of the *Heidelberger Akademie der Wissenschaften* or *Das Deutsche Wörterbuch von Jacob und Wilhelm Grimm auf CD-ROM und im Internet (DBW)* of the Universität Trier, RSNSR uses linguistic as well as statistical rules to represent highly varied spellings. These rules can be automatically derived from evidence data with the possibility of further expert adjustment. This allows a search engine to proceed successfully even for rare spellings without the need of extensive manual operation. A Java-based search engine with a phonetic rule set has already been built [16]. Future versions will be easily integrable into other projects. We are already cooperating with Deutsch Diachron Digital (DDD) [5], which contains texts from Old High German to Modern German. In 2006, after a prototyped solution has been achieved, we plan to integrate the fully functional search engine into the retro-digitization project Nietzsche-CD. In cooperation with UCREL and UCLan we are currently researching the possibilities for a rule-based search engine for Indo-European languages [1].

2 Requirements for hsv-distance measures

One of the main operational points in building a search engine for historical spelling variants is a reliable distance measure. Such a measure can be used in different stages of a query and therefore in more than one module of the engine:

- *Search.* Text retrieval on text in non-standard orthography is obviously more difficult than usual text retrieval. Most standard information retrieval systems build up an index of occurring terms, allowing the user to quickly find all documents containing the words he queried for. As mentioned above, an exact search may not yield good results for historical texts. An adequate distance measure operating on spelling variants provides arbitrary degrees of search fuzziness within a reasonable retrieval time. Standard fuzzy search, though, is of limited use as it does not take linguistic features into account. For example, if the user queries for the German term `urteil` (=judgment), the well known Levenshtein algorithm [14] does not differentiate between the existing variant `urtheil` and, for instance, `ubrteil` with respect to the string distance. A measure that takes heed of linguistic connections will be able to determine the actual variant from a list of candidates.
- *Ranking of Boolean results.* Retrieval in historical text documents is also possible starting from a given query term, using automatically or manually built rules that generate spelling variants. The variants produced are used for Boolean retrieval returning unclassified results. Afterwards, an hsv-distance measure is required to rank the results according to their distance to the term queried.

- *Transformation.* Historical spelling variants should be automatically transformed into their modern counterparts. The hsv-distance measure is used to identify the correct spelling in a dictionary.
- *Reflection.* The differences between a historical or regional spelling variant and its modern equivalent are often hard to evaluate, even for native speakers. An hsv-distance measure is a means of mapping linguistic distinctions on a single number. The visualization of word distances supports the reflection about language as being in a state of constant change.

The amount of support a distance measure can provide depends on its practicability in the particular context of historical spelling variants. Given the abundance of different distance measures and edit-distances available, a thorough evaluation is needed.

3 Comparative study of distance measures

In this section, we briefly describe the measures we compared regarding their retrieval effectiveness: the string edit distance, distances based on an evaluation of n-grams and the Editex algorithm by Zobel and Dart [21], a stochastic distance measure and the new hvs-distance measure computed with our FlexMetric algorithm.

The string edit distance is defined as the minimum number of edit operations needed to transform the one string into the other. These operations consist of character replacements, insertions and deletions. Levenshtein [14] presented a recursive algorithm for calculating the edit distance: Let the function $d(i, j)$ denote the costs needed to transform the first i characters of the string s into the first j characters of the string t. Then the following equations hold obviously:

$$d(0,0) = 0, d(i,0) = i, d(0,j) = j.$$

The complete edit distance for the two strings can then be calculated using the following recursive equation:

$$d(i+1,j+1) = min \begin{pmatrix} d(i+1,j)+1, \\ d(i,j+1)+1, \\ d(i,j)+cost(s_i,t_j) \end{pmatrix}, cost(a,b) = \begin{cases} 0 & \text{if a=b} \\ 1 & \text{otherwise} \end{cases}$$

A more efficient way is to use a dynamic programming approach, as described by Wagner and Fischer [20]. The string edit distance is widely used in a variety of applications as it can be determined efficiently and delivers good results.

Another type of string distance measure relies on the comparison of the n-grams derived from each of the strings. The term n-gram denotes a continuing sequence of n characters. Using padding tokens, $(l + n - 1)$ subsequences can be extracted from a particular string, where l denotes the length of the actual string. For instance, the string 'HISTORICAL' yields the following bigrams:

.H HI IS ST TO OR RI IC CA AL L.

Usually, sets of bigrams or trigrams are compared. There are several possible ways of deriving a non-negative number that represents a distance [6] derived from comparison of the n-gram sets. In our experiments, we used formula 1. In contrast to the other algorithms, it does not denote a distance but a similarity measure for the two strings x and y, where B_x denotes the set of bigrams derived from string x and B_y of string y, respectively:

$$sim(x, y) = 2\frac{|B_x \cap B_y|}{|B_x| + |B_y|} \tag{1}$$

Zobel and Dart [21] presented the Editex algorithm as a new phonetic matching technique. It combines the properties of string edit distances with letter-grouping strategies used in well-known phonetic indexing algorithms like Soundex [13] or Phonix [9]. By doing so, they achieved superior results for tasks of phonetic matching. Basically, it defines an enhancement to the simple string edit distance by introducing a more complex cost function that takes the actual characters being modified into account. Additionally, a double occurrence of characters is implicitly reduced to a single one.

Ristad and Yianilos [19] suggest a stochastic interpretation of string distances. They model them according to the probability of individual operations needed to transform one string into the other. These operations are equivalent to the character replacements, insertions and deletions used to define the string edit distance. Additionally, the probability of identity operations (e.g. `a` to `a`) is taken into account. The actual probabilities are learned from a training set of string pairs using an expectation-maximization algorithm. The authors suggest two different distance measures: the so-called Viterbi distance, which takes into account only the most likely path when transforming the start into the end string, and the stochastic edit distance, which considers all possible paths and also was the one used in our experiments.

The FlexMetric framework developed by one of the authors [12] combines the simplicity of a dynamic programming algorithm with the flexibility of defining arbitrary costs for each possible character transformation.

The basic idea is very similar to the concept behind the string edit distance. The only difference is that, rather than the number of transformations, the costs for the individual operations are taken into account. The costs for the least expensive sequence of operations required to transform the one string into the other define the distance between the two strings. The cheapest sequence can be calculated using a dynamic programming algorithm resembling the one used for evaluating the string edit distance.

As the edit operations correspond with the transformations regarded in the stochastic evaluation previously described, it is possible to derive the actual costs from the probability distribution learned using the expectation-maximization algorithm according to the following principle: The more likely a particular transformation is, the lower the costs that should be assigned to

it. This way, character deviations between modern and historical spellings that occur frequently in the training set lead to cheaper corresponding transformations. Thus, the resulting distance value will also be smaller. The best results are achieved using a logarithmic transformation, as shown in [12].

4 Evaluation methodology

As there are several use cases for a hsv-distance measure and therefore several methods of evaluation, we first describe the assumptions and constraints that lead to solid quality criteria for the particular algorithms. As we concentrate on the effectiveness and not the efficiency of the algorithms, aspects like memory consumption and time needed are not taken into account.

The main problem in judging the quality of string distance measures lies in comparing their applicability for different tasks. It is obvious that a distance measure that has been specifically trained to detect certain linguistic deviations can no longer yield objective results when used to quantify a relation between spellings as it necessarily valuates the trained deviation with lower costs, leading to a shorter distance. Thus, if, for instance, the measure is used to build up a genealogical tree of spelling variants of the same term, it inherently prefers relations it was specifically trained for. This effect leads to unusable results.

In order to avoid this conflict, we have to concentrate on evaluating the potential of the various algorithms for the following text retrieval task: the user queries for the modern spelling, and all documents containing the query term or a historical variant should be returned as results.

Hence, a synthetic information retrieval system (IRS) has to be constructed consisting of a document collection, a retrieval function, and a set of queries along with relevance judgments. This allows the evaluation of the effectiveness of the algorithms with standard methods in Information Retrievalrecall and precision [2].

As we want to concentrate on the algorithms' ability to cognize connections between a query term and its historical spellings, we do not regard a collection of complete texts, but rather a list of words. This way, further factors influencing retrieval results (such as term frequency in the documents) are ignored.

We assembled a list of 3,156 unique pairs of strings, each consisting of a historical deviant spelling and the modern standard spelling. These were manually compiled from 40 historical German documents written from 1350 to 1876. Thus, a number of queries (modern spellings) and relevant answers (historical spellings) for the IRS are found.

The string edit, Editex and FlexMetric distances, can be turned into a normalized similarity function for two strings a, b according to equation 2. The stochastic distance is normalized according to equation 3. These functions yield values between 0 (no similarity / maximum distance) and 1 (identity / no distance). Thus, they can be used to classify the term collection according to the computed similarity to the query in the IRS.

$$sim(a,b) = 1 - \frac{dist(a,b)}{\max\{|a|,|b|\}} \tag{2}$$

$$sim(a,b) = \frac{\min_{c \in Testset} dist(c,b)}{dist(a,b)} \tag{3}$$

To build a collection of searchable terms and spelling variants, we use a manually maintained dictionary of 217,000 contemporary German words derived from the free spelling-correction tool Excalibur. The historical word forms found by the IRS are added to the dictionary, whereas the corresponding modern terms are removed. In this way, it is ensured that no other relevant documents (spelling variants) are in the collection. Hence, we are able to exactly determine the medium recall level after retrieving the first one to five most similar terms and the medium precision level at 100% recall as quality indicators.

If two or more terms are equidistant to the term queried, but just one of them is considered relevant, the worst case is assumed: the sequence of answers is arranged in such a way that the relevant term comes last.

A special problem arises in the case of the stochastic distance measure and the FlexMetric approach as these algorithms require a decent training set of string pairs. In order to maximize the utilization of the manually compiled list, we used cross-validation. The list is randomly split into ten parts of preferably equal length. Nine of them are used to train the distance measures. The newly trained measure is evaluated on the remaining records. This is done ten times, once for each part. The individual results are averaged afterwards.

5 Results and interpretation

Measure	Pr.	R 1	R 2	R 3	R 4	R 5
Bigram evaluation	37.9 %	24.5 %	35.6 %	42.6 %	48.2 %	54.4 %
Editex	56.1 %	43.3 %	55.2 %	63.4 %	69.2 %	72.6 %
Levenshtein	38.9 %	22.9 %	36.6 %	47.1 %	53.4 %	58.9 %
FlexMetric	55.0 %	38.6 %	58.2 %	65.7 %	70.8 %	75.0 %
Stochastic measure	62.4 %	46.7 %	65.3 %	74.7 %	79.6 %	83.1 %

Table 1. Evaluation results

The actual experimental results shown in table 1 can be summarized as follows:

– The string edit distance and n-gram algorithms yield comparable results. This was to be expected as both of them evaluate a deviation regardless of its context or the affected characters respectively.

- The Editex algorithm delivers superior results. It takes into account linguistic aspects due to its letter-grouping strategy. For example, the replacement of a vowel sound with another is in terms of a cost measure cheaper than the replacement of a vowel with a consonant sound. Also, phonetically similar letters are grouped. As our results clearly show, this strategy better reflects linguistic developments than the algorithms that process simple character transformations.
- The results yielded by the stochastic distance and the FlexMetric approach are also above those produced by the basic string edit distance and n-gram algorithms. As they both rely on the same learned probability distribution, this is not surprising. The main difference lies in their conceptual complexity: whereas the stochastic distance measure needs an extensive evaluation of the probability distribution for each term pair, the FlexMetric uses a derived cost measure in a simple dynamic programming algorithm. Hence, it allows intuitive optimizations like re-using previously calculated values for $1 : n$ comparisons. Furthermore, and most important to our field of application, the derived cost measure is more likely to be understood and optimized by a human user, for example, for linguistic analysis.
- In [12], the stochastic and FlexMetric distance delivered precision values of 73.7% and 69.0% respectively. We explain this gain in performance with the nature of the tested set. The evidences evaluated in [12] were compiled from a set of documents originating in a smaller time interval. The advantage of the trainable measures is their ability to adapt to specific features of the training set. Hence, this advantage is lost if the set of documents used for evaluation is compiled from a too broad range of origins and thus contains too many different spelling variants (cf. figure 1).

6 Conclusion

From the results shown above, we draw the following conclusions:

- The better adapted an algorithm is to specific phenomena in the domain of historical spellings, the better the retrieval results that can be expected from it.
- The paramount results of a trained distance measure can be transferred to a simpler evaluation algorithm without significant loss in quality.

In this sense, we have created a simple, easy to handle string distance measure by using a decent training set of string pairs. As an result of our evaluation, this distance measure is capable of correctly identifying unknown historical spelling variants of a given query term with an accuracy of more than 50% and is thus superior to common fuzzy search algorithms like Levenshtein string edit distance or n-gram-based comparisons. We expect a further improvement of the retrieval quality from the usage of a set of trained distance measures: By evaluating a document's metadata, that measure that has been trained on

spelling variants from about the same time interval and location can be used for retrieval. The verification of this assumption is part of our current research.

7 Further work and outlook

The FlexMetric distance measure reflects properties of the spellings it was trained on. Thus, it may be used to detect the occurrence of certain deviations. The fact of their occurrence is, in turn, an indicator of the place and date of the origin of the text. Hence, the FlexMetric can be used to classify texts of unknown origin. Several measures can be trained on text evidence from different times and places. The measure that yields the best results on an unclassified text is assumedly trained on spellings occurring in a text from the same period and location.

Currently, we are developing a collection of trained measures for three time periods between 1350 and 1900 and three German language areas. The evaluation of this approach is part of our research.

The RSNSR project will provide an online search engine that can be used for literature studies by both experts and amateurs. Following the cognitions of a developed prototype, a simplistic interface will be set up. Among its functions is already a visualization of the rules used. An automatic text categorization that estimates the time and location of origin will follow soon.

This engine will then be integrated into different projects in the context of digitizing historical texts. One of these projects will be DDD. The development of our search engine is accompanied by other projects that also provide modules for successful retro-digitization and literature research. Two of these are also held at the Universität Duisburg-Essen: the development of partial text recognition software for German Fraktur fonts [15] and a web-based system for assisted literature research [3]. With these and RSNSR, a framework for the retro-digitization of historical documents is taking shape.

8 Acknowledgements

The presented research was carried out in a recent project Rule-based search in text databases with nonstandard orthography and funded by the German Research Council.

References

1. D. Archer, A. Ernst-Gerlach, S. Kempken S, T. Pilz, and P. Rayson, *The identification of spelling variants in English and German historical texts: manual or automatic?*, proposed for Digital Humanities 2006, July 4 - 9, Paris, France.
2. R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 2000.

3. D. Biella, W. Luther, T. Pilz, *A web-based system for assisted literature research*, Proceedings of the 3rd European Conference on e-Learning 2004, Nov. 25 - 26, Paris, France.
4. Bibliotheca Augustana, FH Augsburg; http://www.fh-augsburg.de/~harsch/augustana.html (accessed 05 Jan. 2006)
5. Deutsch Diachron Digital; http://www.deutschdiachrondigital.designato.de (accessed 05 Jan. 2006)
6. K. Erikson, *Approximate Swedish Name Matching - Survey and Test of Different Algorithms*, Nada report TRITA-NA-E9721, 1997.
7. Excalibur; http://www.eg.bucknell.edu/~excalibr/excalibur.html (accessed 05 Jan. 2006)
8. documentArchiv.de; http://www.documentarchiv.de (accessed 05 Jan. 2006)
9. T. Gadd, PHONIX: The Algorithm, *Program: Automated Library and Information Systems* **24**(4): pp. 363 - 366 (1990).
10. Hessisches Staatsarchiv Darmstadt; http://www.stad.hessen.de/DigitalesArchiv/anfang.html (accessed 05 Jan. 2005)
11. S. Hockey, Living with Google: Perspectives on Humanities Computing and Digital Libraries, *Literary and Linguistic Computing*, **20**: pp. 7 - 24 (2004).
12. S. Kempken, *Bewertung von historischen und regionalen Schreibvarianten mit Hilfe von Abstandsmaßen*, Thesis, Universität Duisburg-Essen (2005).
13. D. Knuth, *The Art of Computer Programming, Vol. 3: Searching and Sorting*, Addison-Wesley, pp. 391 - 392 (1973).
14. V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, *Soviet Physics Doklady*, **10** (8): pp. 707 - 710 (1966).
15. L. Mischke, W. Luther, *Document Image De-Warping Based on Detection of Distorted Text Lines*, in: Fabio Roli, Sergio Vitulano (eds.), *Image Analysis and Processing - ICIAP 2005 proceedings*, Cagliari, Italy, September 2005, LNCS 3617, Springer, pp. 1068 - 1075.
16. T. Pilz, *Unscharfe Suche in Textdatenbanken mit nichtstandardisierter Rechtschreibung am Beispiel von Frakturtexten zur Nietzsche-Rezeption*, Thesis (civil service examination), Universität Duisburg-Essen (2003).
17. T. Pilz, W. Luther, N. Fuhr, U. Ammon, *Rule-based search in text databases with nonstandard orthography*, Proceedings ACH/ALLC 2005, June 15 - 18 Victoria, Canada.
18. P. Rayson, D. Archer, N. Smith, *VARD versus Word: A comparison of the UCREL variant detector and modern spell checkers on English historical corpora*, Proceedings of the Corpus Linguistics 2005 conference, July 14 - 17, Birmingham, UK.
19. E. Ristad, P. Yianilos, Learning String Edit Distance, *IEEE Transactions on Pattern Recognition and Machine Intelligence* **20** (5), pp. 522 - 532 (1998).
20. R. Wagner, J. Fischer, The String-to-String Correction Problem, *Journal of the ACM* **21** (1), pp. 168 - 173 (1974).
21. J. Zobel, P. Dart, *Phonetic String Matching: Lessons from Information Retrieval*, Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 166 - 172 (1996).