Extending a Finite State Approach for Parsing Commas in English to Dutch

Sebastian van Delden and Fernando Gomez

Department of Computer Science, University of Central Florida

Abstract

A finite state approach to determining the syntactic roles of commas has already been established for the English language. Here we extend this approach to Dutch. We identify syntactic dissimilarities of comma usages between the English and Dutch languages and show how much effort is needed to extend this finite state approach to Dutch. Once adapted to the Dutch language, the system is tested across several Dutch sources and results are given.

1 Introduction

Commas are abundant in written texts. A natural language processing system must have a mechanism for identifying the roles that commas play in sentences. Little research, however, has been explicitly devoted towards disambiguating commas.

Recently a finite state approach to determining the syntactic roles of commas has been developed (van Delden and Gomez 2002a), and incorporated into a practical application (van Delden and Gomez to appear, 2002b). This approach combines a set of simple deterministic finite state automata and a greedy learning algorithm to assign descriptive tags to the commas in a sentence—*comma-tagging*. Van Delden and Gomez (2002a) show that the approach achieves 95% accuracy on correctly tagged text. This comma tagging system is a necessary component of any finite state partial parser. If commas are ignored by a partial parser, its output will have many unresolved syntactic issues (for details see van Delden forthcoming). Here we analyze the feasibility of extending this comma-tagging approach to the Dutch natural language, answering the following questions:

1. Are commas used to delimit a similar set of syntax relations in the Dutch language?
2. If a comma-delimited syntactic relation occurs in both English and Dutch, is the syntax of the usage exactly the same?
3. Does this finite state comma-tagging approach perform well on the Dutch language?
4. How much effort is needed to extend this approach to Dutch?

Current part-of-speech tagging strategies (Brants 2000, Zavrel and Daelemans 1999a, Charniak et al. 1996, Brill 1994) do not attempt to characterize commas. The primary reason is the lack of training data—commas are not explicitly tagged in corpora. However, van Delden and Gomez (2002a) show that even if training data is available, a rule-based part-of-speech tagging approach (Brill 1994) performs poorly when trying to determine the syntactic roles of commas.

Our comma-tagging automata rely primarily on part-of-speech information that has already been assigned to each word by a part-of-speech tagger. This supports the extension to other languages, since rule-based part-of-speech taggers which deliver high word-tag accuracy can be trained on and adapted to other languages (Ngai and Florian 2001, Brill 1995).

We define three levels of modification to adapting the English comma-tagging automata to Dutch:

1. No modification at all—the automata can be used as is;
2. Translation of lexicalized arcs;
3. Re-organization of the automata due to the syntactic differences between English and Dutch.

Some syntactic relations can be recognized by the same automata in both English and Dutch because their part-of-speech tag patterns are similar. In such cases, no modification is needed to the English automata that recognize such syntactic relations.

Some automata, however, have been lexicalized to improve performance. Lexicalized automata cannot directly be employed by another language. A simple translation of the lexical term(s) that is (are) assigned to a transition is needed. However, a common problem in machine translation is that one lexical term in a language may result in two or more terms in another language. For example, an automaton is lexicalized to recognize that, if a sentence starts with *For example,...* then *For example* is definitely a transitional phrase that is being concluded by a comma. This phrase, however, would be translated to the single word phrase *Bijvoorbeeld* in Dutch. Simple translation of lexicalized arcs will not always suffice, some arcs may need to be expanded or collapsed.

The ordering of syntactic relations varies greatly in the English and Dutch languages. For example, Dutch prefers time, manner, place as in: *Hij gaat morgen (time) met zijn vrouw (manner) naar Leiden (place).* While English prefers place elements before time elements: *He is going to Leiden tomorrow with his wife.* Verb syntax also varies greatly from English to Dutch when an auxiliary verb or modal is present, for example: *I must go to Leiden tomorrow.* Depending on style, this sentence is translated to Dutch as *Morgen moet ik naar Leiden gaan* or *Ik moet morgen naar Leiden gaan*, which translates directly back to English as *Tomorrow must I to Leiden go* or *I must tomorrow to Leiden go*, respectively. In some cases, an automaton, which captures the new syntactic structure introduced by the Dutch language, must be created to supplement the existing English automaton. In other cases, the English automaton itself must be modified because its syntactic structure does not exist in Dutch.

The remainder of this paper is organized as follows: Section 2 briefly describes the comma tagging approach and compares the comma delimited syntactic relations in the English language to the Dutch language; section 3 shows how the finite state automata should be updated to fit the Dutch language. Section 4 presents an evaluation of the Dutch comma tagging system; and section 5 concludes the paper.

2 Overview of Comma-Tagging

Van Delden and Gomez (2002a) present a algorithm for tagging commas which is divided into two components: 1) A set of simple finite state automata and 2) a co-occurrence matrix. Each automaton is of simple design and runs parallel to the other automata. The result is that a comma may be assigned more than one descriptive tags. Multiple assignments are sometimes necessary to fully disambiguate the comma, for example: *In the Spring of 2003, a great year for football, Tampa Bay won the Super Bowl.* The first comma in this example concludes a prepositional phrase and introduces an apposition. Both roles should be identified.

Because the automata are kept simple and are not ordered, false assignments are commonly made. For example, *I spoke with John, president and CEO of the company*. The comma will be recognized as introducing an apposition, but also as coordinating noun phrases in a list—*John, president and CEO*. The co-occurrence matrix resolves false assignments made by the automata, determining which comma-tags can *co-occur* for a single comma. In short, the matrix is a post automata component which makes it possible to avoid automata ordering and reduce automata complexity. This approach is desirable because van Delden and Gomez (2002a) show that the matrix can be automatically generated from a comma tagged corpus with a greedy learning algorithm.

We have found that this two-step approach to comma tagging is also desirable in Dutch. As in English, a single comma in Dutch can play more than one role. Furthermore, an empirical analysis reveals that co-occurrences in English are almost equivalent to those in Dutch. Co-occurrences are not *exactly* equivalent because some English comma-tags do not exist in Dutch. This phenomenon, however, does not adversely affect the performance of the matrix since the extra co-occurrence information is simply not used in Dutch. Therefore a matrix that is learned from an English corpus can be directly used by a Dutch comma tagger—no conversion work is necessary.

The two-step finite state approach seems to be a viable method for tagging commas in Dutch, but are commas being used to delimit or coordinate a similar set of syntactic relations in the Dutch language? Table 1 shows the set of comma tags that have been defined for the English language. This set was compiled from an empirical analysis of several corpora including the Penn Treebank III (Marcus et al. 1993) and is not meant to be a complete set. Some comma usages occurred so in-frequently that a tag was not warranted. The seventeen possible tag types are divided into 4 categories: series commas, commas used to introduce or enclose non-verbal phrases, commas using to introduce or enclose clauses, and speech commas. The subscript associated with each tag is an indication of the level of

Table 1: Syntactic relations coordinated or delimited by commas in the English language

Group	Tag	Possible Suffixes	Description (coordinates or delimits)
Series	CO-LST$_{1or3}$	-NP$_1$, -PP$_1$, -VC$_3$, -INF$_3$	Series of noun or prep. phrases; verb or infini. clauses
	CO-ADJ$_1$		Two adjectives in a complex noun phrase
Enclosing	CO-APS$_1$	-BEG, -END	Apposition
	CO-PP$_1$	-BEG, -END	Prepositional phrase
	CO-ADV$_2$	-BEG, -END	Adverbial clause
	CO-COR$_1$	-BEG, -END	Coordinated noun or prepositional phrase
	CO-EXP$_2$	-BEG, -END	Explanatory phrase
	CO-DAT$_x$	-BEG, -END	Year part of a date
Clausal	CO-REL$_3$	-BEG, -END	Relative clause
	CO-RREL$_3$	-BEG, -END	Reduced relative clause
	CO-INF$_3$	-BEG, -END	Infinitival clause
	CO-SUB$_3$	-BEG, -END	Subordinate clause
	CO-S$_3$		Independent clause or new sentence
	CO-VC$_3$	-BEG, -END	Verb clause
	CO-RSUB$_3$	-BEG, -END	Reduced subordinate clause
Speech	CO-DIR$_2$		Direct speech
	CO-IDIR$_2$		Indirect Speech

modification needed to extend the associated English comma-tagging automata to Dutch—see section 3 for more details. Suffixes are added to the base tags to identify the type of syntactic relations that are being coordinated, or to indicate whether a delimiting comma is at the beginning or the end of a syntactic relation.

This tag set can be directly used in the Dutch comma tagger, which the exception of the date tag (*CO-DAT*). In English, day follows month in a date and usually has a comma before the year, i.e. *January 15, 1997*. In Dutch, however, month follows day and does not take a comma before the year part, i.e. *15 januari 1997*. The date comma tag therefore does not exist in the Dutch language.

There is also a difference in the types of syntactic relations that are assigned the CO-RSUB and the CO-RREL tags. In English, reduced subordinate and relative clauses are missing the relative pronoun/determiner and auxiliary verb. The introductory verb of the clause is a present or past participle verb. For example: *While walking to school, he met his friend.* or *If opened, the box will explode.* In Dutch, however, there is no present participle verb form—*ik zing* could mean *I sing* or *I am singing* depending on context. A progressive state could also be indicated with a helper verb and the main verb in infinitival form, for example: *He is sleep-*

ing—Hij ligt (lies) te slapen (sleep). Only the second example sentence above can be directly translated to Dutch: *Indien opengemaakt, gaat de doos ontploffen*. The first sentence could be translated as: *Terwijl (while) hij (he) naar school loopt (walk), ontmoette (met) hij zijn vriend*, but would never occur in the reduced form possible in English. The CO-RSUB and CO-RREL tags are, therefore, only assigned to reduced clauses introduced by a past participle verb.

A similar case can be made for appositions. Appositions often occur in the Dutch written language. However, in English, an apposition, that is concluded by a comma, can start a sentence and appositive the noun phrase that follows it. For example: *The best student in the class, John went to the black board. The best student in the class* appositives the noun phrase that follows it, *John*. In Dutch, if a syntactic relation other than the subject of the sentence starts the sentence, then it is usually followed first by the main verb and then the subject of the sentence. In Dutch, it would not sound right to place the subject directly after the apposition. Therefore such sentences are not usually directly translated to Dutch. The apposition is placed after the noun phrase that is being appositived: *John, de bestte student in de klas, ging naar het bord.*

3 Extension to Dutch

The extension of the English comma-tagging automata to Dutch can be divided into three levels of modification: 1) no modification—the FSAs can be used as is; 2) translation of lexicalized arcs; and 3) re-organization due to verb syntax. The following abbreviations are used in the figures that follow in this section:

T	take arc if following tags are present
!T	take arc if following tags are not present
W	take arc if following words are present
!W	take arc if following words are not present
!Comma	take arc if no comma is present
Comma	take arc if comma is present
SOS/EOS	Start/End Of Sentence
RT	set of relative determiner/pronoun tags
VP	any type of verb tag
NP	any type of noun tag
ADV	any type of adverb tag
AUX	auxiliary verb
MD	modal verb
PP	any preposition tag
VBN	past participle verb tag
REL	relative clause grouping (pre-processing)
INF	infinitival clause grouping (pre-processing)
CC	coordinate conjunction tag

3.1 No Modification

Some of the comma tagging automata that were developed for the English language can be directly used without modification. The comma-tags assigned by these automata are used to delimit or coordinate: lists of noun or prepositional phrases (CO-LST-NP or CO-LST-PP), adjectives (CO-ADJ), appositions (CO-APS), prepositional phrases (CO-PP), coordinated noun or prepositional phrases (CO-COR).

These automata are not lexicalized and do not contain verb clauses, For example, an automaton designed to recognize a list of noun phrases will do so based on part-of-speech information, regardless of the language: *I must speak with* ***my parents, the teacher and the director of the school*** *at once.—Ik moet meteen met* ***mijn ouders, de leraar en de directeur van de school*** *praten.* Even though syntactic order varies dramatically in this sentence from English to Dutch, the simple FSA shown in figure 1 will tag both of these commas correctly. Note that the self arc on node D may contain infinitival or relative clause groupings identified by pre-processing automata (See section 3.3 for more details). Similar cases can also be made for the other automata that do not need modification.

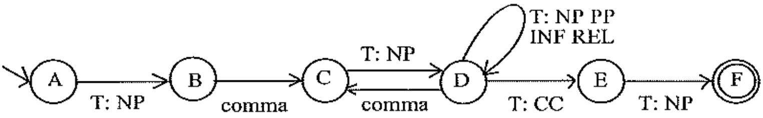

Figure 1: Finite state automaton which recognizes commas that coordinate a list of noun phrases in both English and Dutch.

3.2 Translation of Lexicalized Arcs

Some of the comma tagging automata that have been lexicalized cannot be directly applied to the Dutch language without some minor modifications. Four types of tags belong to this group: adverbial phrases (CO-ADV), explanatory phrases (CO-EXP), indirect (CO-IDIR) and direct speech (CO-DIR). Extending the automata which assign these tags to Dutch, simply calls for a translation of the lexicalized transitions, adding or removing arcs when necessary. For example, figure 2 shows the two automata which identify adverbial phrases delimited by commas.

The first automaton (top) identifies commas which enclose a sequence of adverbs, for example: *Furthermore,...* or *Bovendien,....* This automaton relies solely on part-of-speech information and applies to both English and Dutch without modification. The second automaton in figure 2 (bottom), however, identifies the comma that concludes an adverbial noun phrase which starts a sentence. For example, *Two weeks ago,...* or *Twee weken geleden,....* The arc B-C is taken only if a particular *time* or *location* word is at that position in the sentence. In

English these words include: today, hour(s), day(s), week(s), month(s), year(s), north, south, etc. To apply this automaton to Dutch, these words must be translated: vandaag, uur (uren), dag(en), week (weken), maand(en), jaar (jaren), noord, zuid, etc. A similar case can be made for the remaining syntactic relations in this section.

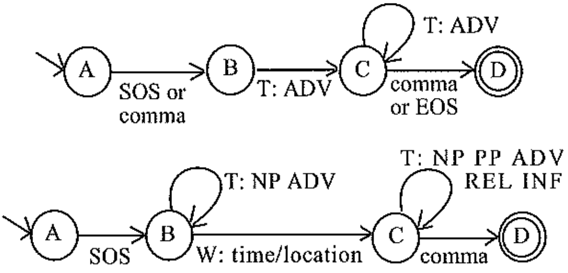

Figure 2: Adverbial phrase automata. Top automaton recognizes commas that enclose a sequence of adverbs in Dutch and English. Bottom automaton recognizes the comma that concludes adverbial noun phrases and requires translation for Dutch.

3.3 Syntactic Re-Organization

Most complications result from the differences of verb syntax in English and Dutch. In these cases, a re-organization of the automata is necessary for classification to be accurate. The comma tags affected here delimit or coordinate the following clauses: pre-processed relative clauses (REL), lists of verb and infinitival clauses (CO-LST-VC and CO-LST-INF), verb clauses (CO-VC), relative clauses (CO-REL), subordinate clauses (CO-SUB), infinitival clauses (CO-INF), and independent clauses (CO-S). New automata need to be defined by re-arranging the arcs and possibly adding new states and transitions. Some of the English automata are still used, but require an extra automaton to handle the new possible syntax introduced by Dutch.

In the English comma-tagger, a pre-processing step groups all words up to and including the introductory verb phrase of relative or infinitival clauses that are not delimited by commas. These groupings are assigned a single tag (INF or REL) to be used by the comma tagging automata that follow. Special pre-processing automata perform these groupings which is also a necessary step in the Dutch comma tagger. For example, *In her haste to leave the store, Emma forgot her purse.—In haar haast de winkel te verlaten, vergeet Emma haar portemonnaie. Te verlaten* is grouped together and assigned a single tag (INF), reducing the complexity of the automaton that will identify the comma as concluding the prepositional phrase *In haar haast de winkel te verlaten.* These pre-processing automata help simplify all of the remaining comma tagging automata.

Infinitival clause groupings can be accomplished directly by the English infinitival clause pre-processing automaton. The relative clause pre-processing automaton, however, requires an extra complication due to the different ordering of verb complements in English and Dutch. For simple past and present tense where no auxiliary or modal is present, the English relative clause pre-processing automata will work in the Dutch language. For example, *I bought a radio, a television that I* ***returned*** *this morning and a video machine.—Ik kocht een radio, een televisie die ik vanmorgen* ***terugbracht*** *en een video.* The first verb phrase after the relative determiner is identified as the introductory verb phrase of the relative clause. However, if an auxiliary verb or a modal is present, the verb phrase can be divided by post verbal noun or prepositional phrases. Consider the following example, *I bought a radio, a television that I* ***must return*** *to the store and a video machine.—Ik kocht een radio, een televisie die ik* ***terug*** *naar de winkel* ***moet brengenen*** *een video.* Note that in Dutch the prepositional phrase *naar de winkel* can be placed in between the modal and the main verb. The relative clause pre-processing automaton must be extended to handle this possibility. In English, adverbs are commonly placed in between an auxiliary or modal and the main verb, but very seldom noun or prepositional phrases, so this extra complication is not needed in English. Figure 3 shows the relative clause pre-processing automaton in English (top) and the new automaton that is added for Dutch (bottom).

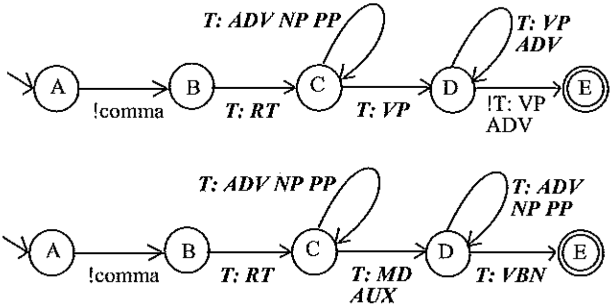

Figure 3: The pre-processing relative clause automata. The top automaton is used for the English and Dutch languages. The bottom automaton is used only by the Dutch language.

The highlighted/italicized arcs indicate the words that are being grouped together. An extra automaton is added instead of complicating the original one. Both automata are used in Dutch. When applying these automata, the new automata in figure 3 (bottom) would be applied prior to the old one (top). If applied in the opposite order, the old automaton would still accept, inhibiting the new one. To revisit the previous example, the following words would be grouped by this pre-processing automaton: *Ik kocht een radio, een televisie (die ik* ***terug*** *naar de* *winkel* ***moet brengen)/*** *REL en een video machine.* The simple automaton in figure 1 can now recognize that the comma coordinates a list of noun phrases in this sentence.

The automata that recognize commas enclosing relative and subordinate clauses also need modification. Figure 4 shows the original automata used in English to recognize commas that enclose relative clauses and the new set needed for Dutch.

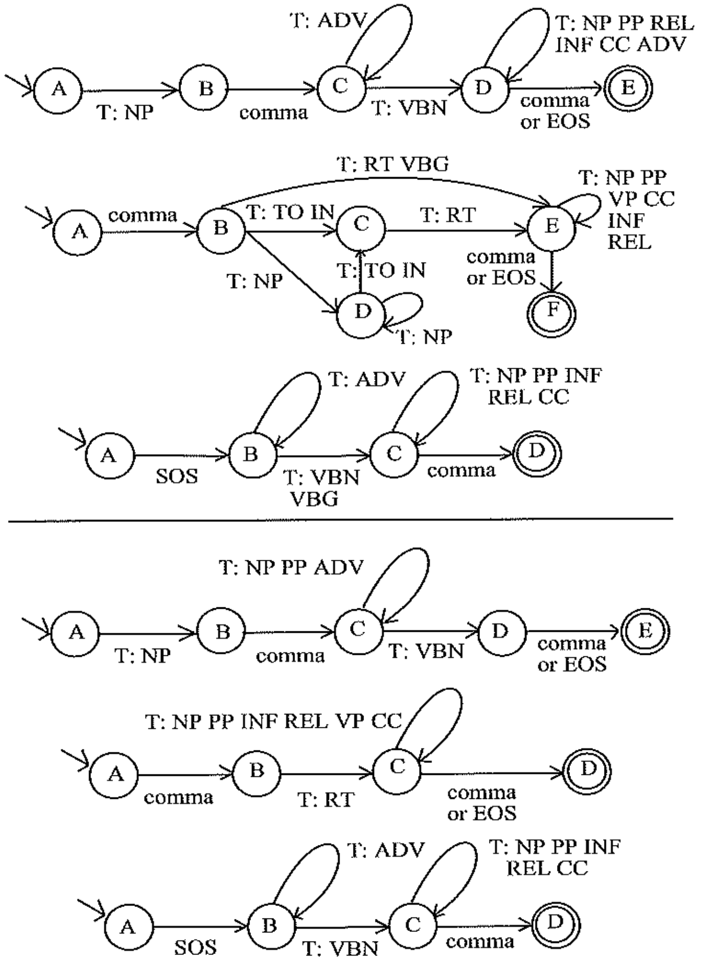

Figure 4: Relative clause automata. The automata on the top are used in English, and the automata below are used in Dutch.

The three automata in the upper half of figure 4 are used to tag commas enclosing relative clauses in English. The first recognizes commas that enclose reduced relative clauses with a past participle introductory verb. The second recognizes commas that enclosed relative clauses introduced with relative determiners or pronouns, for example *I read 10 books, which were all interesting.* or *I read 10 books, of which the first was interesting.* or *I read 10 books, the first of which was interesting.* The third automaton in the upper half identifies the comma that concludes a reduced relative clause that starts a sentence.

The first automaton in the upper half is still used in Dutch, and is supplemented by the first automaton in the lower half. This new automaton also identifies commas that enclose a reduced relative, but the past participle verb is located at the end of the clause—which does not occur in English. For example: *The method, **also called** smelting, takes 2 hours.—De methode, **ook** smelting **genoemd**, duurt 2 uren.*

The second automaton in the lower half replaces its English counterpart because, in Dutch, prepositional compounds replace prepositions plus relative pronouns or determiners. For example, *of which—waarvan, in which—waarin, with which—waarmee, upon which—waarop*, etc. These prepositional compounds simplify the implementation of the second automaton while preserving its capabilities: *I read 10 books, **the first of which** was interesting.—I las 10 boeken, **waarvan het eerste** interessant was.* Note that this automaton is rather relaxed and still recognizes the commas if the verb phrase is separated, as in: *The books, that I **had given** to Jan, are now lost.—De boeken, die ik aan Jan **had gegeven**, zijn nu verloren.*

Finally, the last automaton in the lower half of figure 4 recognizes a comma that concludes a reduced relative clause that starts a sentence. The only difference with this automaton and the English version is that only past participle verbs can introduce the clause, as in: *Tired from walking, Piet went to sleep.—Vermoeid van het wandelen, ging Piet slapen.*

Lists of verb clauses can be syntactically similar in English and Dutch, as in: *I **kicked** the ball to Jan, **ran** left towards the goal and **waited** a few minutes.—Ik **schopte** de ball naar Jan, **rende** links naar de goal en **wachtte** voor een paar minuten.* However, because of the new verb syntax capabilities in Dutch, additional automata must also be defined here. For example: ***I must read** the book, **sell** my bicycle and **find** my notebook.—Ik **moet** het boek **lezen**, mijn fietsverkopen en mijn schrift **vinden**.* If a modal or auxiliary verb is present, the main verb appears at the *end* of each verb clause in the sentence.

Furthermore, it needs to be noted that in Dutch a list of verb clauses can be very similar to a list of infinitival clauses. In English, *to* always precedes an infinitival clause, making it easier to distinguish an infinitival verb from a main verb. In Dutch, this distinction is not made in a list of infinitival clauses, for example: *I **want to read** the book, **sell** my bicycle and **find** my notebook.—Ik **wil** het boek **lezen**, mijn fiets **verkopen** en mijn schrift **vinden**.* Note that this Dutch sentence is syntactically almost equivalent to the previous one above—the only difference is the use of an auxiliary verb instead of a modal. If a modal is present, then this is a list of verb clauses, otherwise if there is an auxiliary verb present, this is a list of infinitival clauses.

Similar to lists of verb clauses, when a single verb clause is being coordinated by a comma and a conjunction, the syntax can also differ dramatically from English to Dutch when a modal or auxiliary verb is present. In English the introductory verb phrase is always at the beginning of the clause. In Dutch, however, it is possible for the main verb to be located at the end: *I* ***have read*** *the book, and* ***seen*** *the movie.—Ik* ***heb*** *het boek* ***gelezen****, en de film* ***gezien****.*

Purpose infinitival clauses are also written differently in Dutch as compared to English. The place-holder word *om* which translates to *in order to* is almost always used in Dutch. It is followed by post verbal noun and prepositional phrases and finally the infinitive. For example: ***To climb*** *the mountain, John must first buy good shoes.—Om de berg* ***te beklimmen****, moet John eerst goede schoenen kopen.* As with the relative clause automata, changes must be made to all of the automata in this section in order to capture the new syntactic structures possible in Dutch.

4 Evaluation

Brill's rule-based part-of-speech tagger was used to assigned part-of-speech tags to the words in the test sentences. The tagger had been trained on a section of the Eindhoven Corpus (uit den Boogaart 1975) by Edwin Drenth and was available for download. The tagset used is similar to the WOTAN and WOTAN-II tagsets (van Halteren 1999, Zavrel and Daelemans 1999a). The tags were translated to Penn Treebank Tag Set (Santorini 1995, Marcus et al. 1993) which is used by the arcs of the automata in this approach.

The comma tagging system was tested on the same section of the Eindhoven corpus, used by Edwin Drenth to train Brill's tagger, and on random articles taken from Rotterdam's online newspaper and three online encyclopedias:

- Sterrenkunde `http://www.astro.uva.nl/encyclopedie`
- Gezondheid `http://www.gezondstegids.nl`
- Eletrotechniek `http://www2.ele.tue.nl/encyclopedie`

No changes were made to the automata once testing began. Results are shown in table 2. Improper usage of commas in sentences were not included in the analysis. Interestingly, we found incorrect placement of commas to be similar in the English and Dutch languages. For example, *The tree that had stood for over a hundred years, was blown over by the hurricane.—De boom die voor honderd jaar stond, was omgewaaid door de orkaan.* In this example, a comma is used to conclude a relative clause but there is no introductory comma. This comma is therefore classified as being incorrectly used and would have not been included in the analysis.

Overall, the Dutch system performed about the same as the English version of the system. Typical errors that are made due to the limitations of the automata when tested in English (van Delden and Gomez 2002a) were also made here. For

Table 2: Dutch comma tagging results

Source	**Avg. Sentence Length**	**Number Commas**	**Accuracy**
Eindhoven Corpus	20	6246	94.5%
S-Encyclopedia	27	244	93.4%
G-Encyclopedia	17	72	93.5%
Rotterdam Newspaper	22	62	95.2%
E-Encyclopedia	17	51	92.0%

example, consider the following sentence that was encountered during testing: *For stream 1 you use, for example, light green, and for stream 2 red.—Voor stroom 1 gebruik je bijvoorbeeld groen licht, en voor stroom 2 rood.* The system would identify the last comma as coordinating a prepositional phrase, when actually an independent clause containing an elliptical structure is being coordinated: *and for stream 2* ***you use*** *red—en voor stroom 2* ***gebruik je*** *rood.* A detailed analysis of the entire sentence is needed to recognize elliptical structures and is beyond the scope of this finite state approach.

A new source of ambiguity was, however, introduced by the Dutch language that is not encountered in English—distinguishing between certain coordinated verb clauses and independent clauses. Consider the following sentence that was encountered during testing: *Opposing loads pull at each other, and so electric forces* ***hold*** *the whole world together.—Tegegestelde ladingen trekken elkaar aan, en zo* ***houden*** *elektrische krachten de hele wereld bij elkaar.* In this sentence, an independent clause is being introduced by a comma and a conjunction. Notice that in Dutch the verb is placed before the subject in the independent clause—*houden elektrische krachten.* This creates an ambiguity when a verb clause is being introduced and not an independent clause, for example: *Opposing loads pull at each other, and so* ***hold*** *the whole world together.—Tegengestelde ladingen trekken elkaar aan, en* ***houden*** *zo de hele wereld bij elkaar.* Verb sub-categorization knowledge must be considered to realize that the verb *houden* (hold) is transitive and cannot take two noun phrase objects. Unlike the WOTAN tagsets, however, the Penn Treebank tagset used here does not include this information. Had the WOTAN tagset been used in our system, the automata could have been modified to more accurately handle such situations.

Finally, we conclude this section by listing some example sentences encountered during the evaluation to illustrate the complexity of the sentences that can be corectly handle by this approach (The English translations here illustrate how the commas are being used and attempt to preserve the Dutch syntax).

(1) *The presence began to be noticed even outside of the building,/CO-PP-BEG in order words through the antennas on the roof,/CO-LST-NP the call signs and the name of the club in front of the windows,/CO-PP-END|CO-COR-*

BEG and through the longwire-antenna that runs from the roof of E-High to E-Low.
Zelfs buiten het gebouw valt de aanwezigheid al te merken,/CO-PP-BEG onder andere door de antennes op het dak,/CO-LST-NP de roepletters en de naam van de club voor de ramen,/CO-PP-END|CO-COR-BEG en door de langdraad-antenne die van het dak van E-Hoog naar E-Laag loopt.

(2) *That is immediately more friendly to the environment,/CO-SUB-BEG because satalites remain,/CO-SUB-BEG after they have served their purpose,/CO-SUB-END as rubbish in Space.*
Dat is meteen milieuvriendlijker,/CO-SUB-BEG want satellieten blijven,/CO-SUB-BEG nadat ze dienst hedden gedaan,/CO-SUB-END als afval in the ruimte achter.

(3) *If you pull your jacket over your recently washed,/CO-ADJ dry hair,/CO-SUB-END you can even see the sparks right in front of your eyes.*
Als je je trui over je net gewassen,/CO-ADJ droge haren trekt,/CO-SUB-END kun je de vonkjes zelfs vlak voor je ogen zien.

(4) *From the moment of this "Big-Bang",/CO-PP-END|CO-REL-BEG the origins of which we still do not precisely know,/CO-REG-END the universe started to expand and cool off,/CO-PP-BEG to the condition that we are currently in.*
Vanaf het moment van deze "oerexplosie",/CO-PP-END|CO-REL-BEG waarvan men nu nog niet weet hoe deze precies heeft plaatsgevonden,/CO-REL-END is the heelal gaan uitdijen and dus gaan afkoelen,/CO-PP-BEG tot de toestand waarin wij het nu zien.

5 Conclusions

We have shown how a two-step finite state approach to determining the syntactic roles of commas can be extended to the Dutch language. At the first step, three levels of modification are needed to covert English comma-tagging automata to Dutch: 1) No modification—the automata can be used as in; 2) Because some automata have been lexicalized, a simple translation of lexicalized transitions will suffice; and 3) Primarily due to different possible verb syntax capabilities in Dutch, some automata must be re-designed or supplemented with an additional automaton. At the second step, a co-occurrence matrix that has been generated from an English corpus can be directly used by the Dutch comma tagger—no conversion work is necessary.

Results obtained from testing the system on several Dutch corpora are similar to those previously found in English—about 92-95% on correctly tagged texts. As in English, a Dutch comma tagger can play a crucial intermediate role in a natural language processing system, reducing the overall complexity of the system and resolving crucial syntactic issues.

References

uit den Boogaart, P. (1975), *Woordfrequenties in Geschreven en Gesproken Nederlands*, Oosthoek, Scheltema en Holkema, Utrecht.

Brants, T. (2000), A statistical part-of-speech tagger, *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, Seattle, pp. 224–231.

Brill, E. (1994), A report of recent progress in transformation-based error-driven learning, *Proceedings of the ARPA Human Language Technology Workshop*, Princeton.

Brill, E. (1995), Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, *Computational Linguistics* **21** (4), 543–565.

Charniak, E., Carroll, G., Adcock, J., Cassandra, C., Gotoh, Y., Katz, J., Littman, M. and McCann, J. (1996), Taggers for parsers, *Journal of Artificial Intelligence* **85** (1–2), 45–57.

van Delden, S. (forthcoming), *Larger First Partial Parsing*, PhD thesis, Computer Science, University of Central Florida, Orlando.

van Delden, S. and Gomez, F. (2002a), Combining finite state automata and a greedy learning algorithm to determining the syntactic roles of commas, *Proceedings of the 14th International IEEE Conference on Tools with Artificial Intelligence*, Washington, DC, pp. 293–301.

van Delden, S. and Gomez, F. (2002b), Retrieving NASA problem reports with natural language, *7th Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*, Stockholm, pp. 150–159.

van Delden, S. and Gomez, F. (to appear), Retrieving NASA problem reports: A case study in natural language information retrieval, *Journal of Data and Knowledge Engineering*.

van Halteren, H. (ed.) (1999), *Syntactic Wordclass Tagging*, Kluwer Academic Publishers, Dordrecht.

Marcus, M., Santorini, B. and Marcinkiewicz, M. (1993), Building a large annotated corpus of English: The Penn treebank, *Computational Linguistics* **19** (2), 313–330.

Ngai, G. and Florian, R. (2001), Transformation-based learning in the fast lane, *Proceedings of the NAACL 2001 Conference*, Pittsburgh, pp. 40–47.

Santorini, B. (1995), Part-of-speech tagging guidelines for the Penn treebank project, *Technical report*, Department of Computer Science, University of Pennsylvania. 3rd revision, 2nd printing.

Stern, H. (1984), *Essential Dutch Grammar*, Dover Publications Inc., New York.

Zavrel, J. and Daelemans, W. (1999a), Evaluatie van part-of-speech taggers vor het Corpus Gesproken Nederlands, *Technical report*, CGN-Corpusannotatie.

Zavrel, J. and Daelemans, W. (1999b), Recent advances in memory-based part-of-speech tagging, *Actas del VI Simposio International de Comunicacion Social*, Santiago de Cuba, pp. 590–597.