KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 89–95

CHAIN ALGORITHM USED FOR PART OF SPEECH RECOGNITION

ANDREEA-DIANA MIHIS[1]

Abstract. Dictionary base methods have the advantage that they can be applied to texts written in different languages if there exist an electronically dictionary for that specific language. As a consequence, these kinds of methods can be used to identify the part of speech of words from a text written in a single language. The principal advantage of using a dictionary base approach in part of speech recognition is that it can be applied to different languages, because it does not use a specific grammar. In this paper such a method is used to identify the parts of speech of words from a text using a chain algorithm [7] that disambiguates a text.

1. Introduction

When learning a language, the use of a dictionary is essential, even though grammar skills are not fully developed. When translating a text from a foreign language, the first step consists in identifying the corresponding word, followed by applying mostly grammar skills of the familiar language in trying to obtain the correct translation. When translating from a familiar language to a foreign language the most used method is word by word (the french mot-a-mot). From beginners point of view the dictionary is a powerfull tool, that sometimes can be used successfully to overcome most dificulties. This should be appliable to a computer. Like the beginner, the computer can use the dictionary explanation to choose the correct form of a word and to decide their speech part in a text. This method can be used to many languages, as long as there is an appropiate electronical dictionary.

Pertaining to Natural language processing, this method uses only a dictionary to identify the part of speech of a given word. The basic idea is that in the dictionary, every word will have different definitions, one or more for every part of speech of the specified word. Trying to identify the correct part of speech from

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.
Key words and phrases. chain algorithm, part of speech recognition.

the list, is almost equivalent with trying to match the definitions. And matching the definitions is done using the Chain Algorithm.

2. Chain algorithm for part of speech recognition

The chain algorithm for part of speech recognition is based on and includes the chain algorithm presented in [7].

The idea of the algorithm is to try to disambiguate all the words, in groups of three, but, because the part of speech (POS) is unknown, the word is searched in dictionary with all POS(s). If a word is not found with a POS, then the senses of that word for that POS does not exists. Finally, the POS and the sense merge from the greatest score.

The algorithm for POS recognition by disambiguating a triplet of words $w_1w_2w_3$ for Dice measure is the following:

begin
  for each POS p^{w_1} do
    for each POS r^{w_2} do
      for each POS q^{w_3} do
        begin
          for each sense $s^i_{p^{w_1}}$ do
            for each sense $s^j_{r^{w_2}}$ do
              for each sense $s^k_{q^{w_3}}$ do
                $score(i,j,k) = 3 \times \frac{|D_{p^{w_1}} \cap D_{r^{w_2}} \cap D_{q^{w_3}}|}{|D_{p^{w_1}}|+|D_{r^{w_2}}|+|D_{q^{w_3}}|}$
              endfor
            endfor
          endfor
        end
      endfor
    endfor
  endfor
  $(P, R, Q, i^*, j^*, k^*) = argmax_{(p,r,q,i,j,k)} score(i,j,k)$
  /* POS of w_1 is P, sense of P^{w_1} is $s^{i^*}_{P^{w_1}}$
  POS of w_2 is R, sense of R^{w_2} is $s^{j^*}_{R^{w_2}}$
  POS of w_3 is Q, sense of Q^{w_3} is $s^{k^*}_{Q^{w_3}}$ */
end

Here, α^w denotes the part of speech α of word w, and $s^i_{\alpha^w}$ is the sense with number i for the word w with POS α. D_{α^w} is the complete dictionary definition for word w with POS α.

For the overlap measure the score is calculated as:

$score(i,j,k) = \frac{|D_{p^{w_1}} \cap D_{r^{w_2}} \cap D_{q^{w_3}}|}{min(|D_{p^{w_1}}|,|D_{r^{w_2}}|,|D_{q^{w_3}}|)}$

For the Jaccard measure the score is calculated as:

$score(i,j,k) = \frac{|D_{p^{w_1}} \cap D_{r^{w_2}} \cap D_{q^{w_3}}|}{|D_{p^{w_1}}| \cup |D_{r^{w_2}}| \cup |D_{q^{w_3}}|}$

The POS recognition algorithm is a chain algorithm. This means that for the first triplet it is used to identify the POS(s) of all tree words, and for the following ones, only for the last word, because the POS(s) of first two are already identified. As a consequence, the complexity of the algorithm decreases:

begin
  for each POS q^{w_3} do
    begin
      for each sense $s^k_{q^{w_3}}$ do
        $score(i*, j*, k) = 3 \times \frac{|D_{P^{w_1}} \cap D_{R^{w_2}} \cap D_{q^{w_3}}|}{|D_{P^{w_1}}| + |D_{R^{w_2}}| + |D_{q^{w_3}}|}$
      endfor
    end
  endfor
$(P, R, Q, i^*, j^*, k^*) = argmax_{(p,r,q,i,j,k)} score(i,j,k)$
/*POS of w_3 is Q, sense of Q^{w_3} is $s^{k^*}_{Q^{w_3}}$ */
end

As it can be seen in detail in [7], some "stop" words were used to eliminate the trivial cases of glosses intersection.

3. Experimental evaluation

To test the POS recognition algorithm were used ten files from Brown corpus thanks to the possibility offered by SemCor corpus to evaluate the results. The input files were not tagged, and they contain compound words, underscored.

The algorithm was tested only for English Language, having WordNet as base dictionary, implying the following POS(s): Nouns, Verbs, Adjectives and Adverbs.

The application that was implemented to test the algorithm has as output the list of words tagged with theirs POS(s) and theirs corresponding sense, as follows: *noun*#*n*#*i*. Here the noun *noun* has the POS noun and the sense i from WordNet. If #..# were missing, than the algorithm failed to find the POS and sense of the corresponding word.

Because in SemCor exist some words without tag (Notag) in computing the precision, they and the stop words are ignored.

The final results can be seen in Figure 1 and in detail in annex.

In the upper part of the figure is displayed the precision for the whole file, ascending. The precision for every part of speech is displayed in the lower part of the figure, using the same order as for the first one. The precision for a specific POS is computed thus if a word has a specific POS, it will be recognized with the precision found in the graph. There are some words that are identified to have a specific POS, even if in that case they have another POS, but these words are not considered in computing the precision shown in the Figure 1.

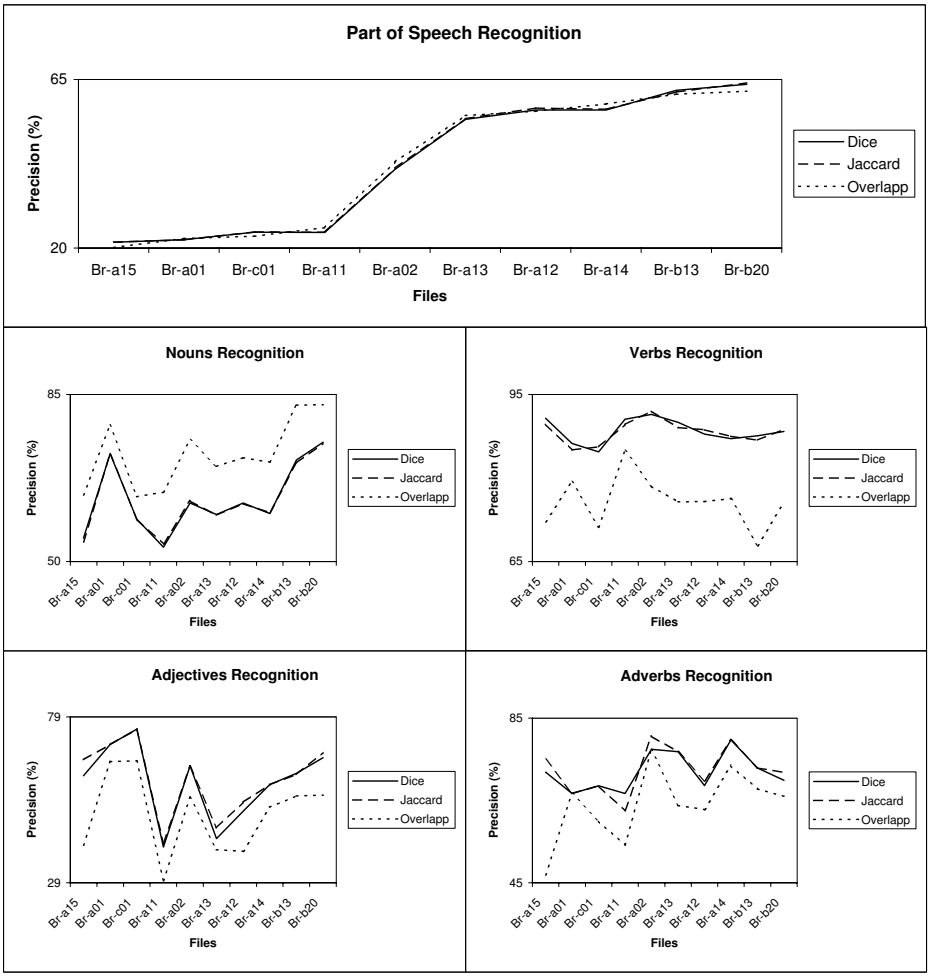

FIGURE 1. Precision of POS Recognition sorted ascending by the global precision

Because not all POS(s) were considered in applying the POS recognition algorithm, the precision for the whole file is lower then the precision for every POS. It seems that the highest precision is for verbs and the lowest one is for adjectives. In case of Nouns, the highest results were obtained using Overlapp measure, but in the other cases, with Overlapp measure the lowest results were obtained.

Table 1 summarizes the results.

Precision	Minimum Value	Maximum Value	Average Value
POS Recognition	20.16	64.13	42.78
Disambiguation	7.00	41.43	21.78
Nouns	53.04	82.83	66.12
Verbs	67.83	92.02	84.45
Adjectives	29.55	75.34	56.39
Adverbs	46.87	80.65	69.41

TABLE 1. Minimum, Maximum and Average Value for Precision

The Precision for disambiguating the whole text is not great, only 21,78% in average. But the precision for a specific POS is by average more then 68%.

4. Application of POS recognition

An application of POS recognition algorithm is in Entity-Relation diagrams automatic construction.

To create an Entity-Relation diagram from a specification written in a specified natural language, first step is to identify the nouns - Entities and verbs - Relations.

For instance, for specification: *A driver drives cars* the following result is obtained:

$Word : driver\ Driver\#n\#1 : driver$

$Word : drives\ Drives\#v\#28 : drive, take$

$Word : cars\ Cars\#n\#2 : car, railcar, railwaycar, railroadcar$

The corresponding Entity-Relation Diagram will have as Entities nouns: *driver* and *car*, and as Relation the verb *drives*.

Next step is to identify Attributes for the Entities. This can be done by searching for the adjectives which determine the noun, and by searching for the nouns bounded to the Entity noun with an membership verb, as in the following: $APeoplehasanAddress$. *Address* is an Attribute for the Entity *Pupil*.

5. A short compression to another related methods

Most of the methods found in the literature that deals with part of speech identification, or word tagging, are designed only for a single language. They use grammatical notions, corpuses and artificial intelligence techniques such as training on a part of the target text. They have very good results: more than 90% precision.

The advantage of the proposed method is that it is not using grammatical concepts, and so, it can be used for more than one language. And because of this, if a bilingual dictionary is available, the algorithm can be used to identify the language of every word.

6. Conclusion and further work

POS recognition algorithm can be used to identify the POS(s) of words from a text written in a specified language if there is an electronically dictionary, without using grammar notions or another sources.

The precision for POS recognition can be improved. For instance, if at the beginning all the words with only one POS and only one sense are annotated, and than the CHAIN algorithm is applied ascending and descending, using the words with only one POS and sense as anchors. Another way is by not ignorring some POS(s), and this can be easily done by considering all the POS(s) found in the dictionary for the corresponding language. Another posibility is to start with the word with the longest gloss. And of course, combining the Chain algorithm with an artificial intelligence technique, to maximize the sum of all scores.

The POS recognition algorithm is a consequence of CHAIN algorithm for word sense disambiguation found in [7]. I believe that the CHAIN algorithm can have some more interesting consequences.

7. Annex

See Figure 2.

8. References

[1] S. Banarjee and T. Pedersen. 2003. *"Extended Gloss Overlaps as a Measure of Semantic Relatedness."* Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, August 9-15, Acapulco, Mexico.

[2] E. Agirre and P. Edmonds (editors). 2006. *"WSD: Algorithms and Applications."* Springer.

[3] C. Fellbaum (editor). 1998. *"WordNet An Electronic Lexical Database."* The MIT Press.

[4] D. Jurafsky and J. Martin. 2000. *"Speech and language processing."* Prentice Hall.

[5] C. Manning and H. Schutze. 1999. *"Foundation of statistical natural language processing."* MIT.

[6] D. Tatar and G. Serban. 2001. *"A new algorithm for WSD."* Studia Univ. Babes-Bolyai, Informatica, 2, 99108.

[7] D. Tatar, G. Serban, A. Mihis, M. Lupea and M. Frentiu. *"A chain dictionary method for Word Sense Disambiguation and applications"*, Proceedings of KEPT2007, to appear.

[8] http://wordnet.princeton.edu/perl/webwn

[1] Computer Science Department, Babes-Bolyai University, Kogalniceanu Street nr. 1, RO-400084, Cluj-Napoca, Romania

E-mail address: mihis@nessie.cs.ubbcluj.ro

	Br-a01			Br-a02			Br-a11			Br-a12			Br-a13			Br-a14			Br-a15			Br-b13			Br-b20			Br-c01		
	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.
Number of Words:	1816	1816	1816	1765	1765	1765	1843	1843	1843	1922	1922	1922	1856	1856	1856	1844	1844	1844	1908	1908	1908	1881	1881	1881	1860	1860	1860	1967	1967	1967
Correct Disambiguated:	128	128	193	217	217	342	117	118	211	319	316	502	289	295	479	287	281	456	90	94	176	334	334	514	339	338	500	128	127	191
Correct POS:	280	279	284	468	492	512	304	305	320	731	738	727	665	668	678	663	665	682	277	278	259	784	778	771	769	774	747	312	313	298
Not in WordNet:	30	30	30	61	61	61	37	37	37	110	110	110	75	75	75	99	99	99	69	69	69	93	93	93	130	130	130	77	77	77
Nouns:	215	215	408	408	408	296	296	296	296	570	570	570	578	578	578	541	541	541	233	233	233	505	505	505	460	460	460	236	236	236
Correct nouns:	156	156	169	254	256	308	157	159	191	355	354	409	346	345	404	325	326	383	128	126	149	360	357	418	345	344	381	138	139	150
False nouns:	31	32	43	59	60	99	69	70	99	131	128	213	130	127	204	125	124	187	42	42	88	121	120	210	107	102	187	37	37	75
Verbs:	87	87	163	163	163	106	256	256	256	256	256	256	210	210	210	232	232	232	97	97	97	258	258	258	258	258	258	111	111	111
Correct verbs:	75	74	69	149	150	128	96	95	90	225	227	194	189	187	159	202	203	177	88	87	70	226	224	175	228	229	195	94	95	79
False verbs:	45	45	29	101	98	50	71	71	44	123	123	62	123	124	63	130	131	65	53	55	30	152	155	93	102	102	63	50	50	31
Adjectives:	58	58	58	95	95	95	88	88	88	182	182	182	182	182	182	145	145	145	62	62	62	250	250	250	186	186	186	73	73	73
Correct adjectives:	41	41	38	61	61	52	35	36	26	92	97	70	77	83	71	85	85	75	38	41	25	155	154	138	124	127	103	55	55	48
False adjectives:	9	9	11	21	18	13	26	25	16	50	47	46	45	45	37	45	42	38	9	8	15	30	34	21	52	50	42	28	27	23
Adverbs:	12	12	12	31	31	31	24	24	24	86	86	86	69	69	69	64	64	64	32	32	32	59	59	59	103	103	103	35	35	35
Correct adverbs:	8	8	8	24	25	24	16	15	13	59	60	54	53	53	44	51	51	47	23	24	15	43	43	40	72	74	68	24	24	21
False adverbs:	4	4	2	15	16	10	26	25	17	35	34	22	44	43	25	27	28	18	20	18	9	32	32	24	18	20	9	2	2	2
Ignored:	528	528	528	514	514	514	548	548	548	526	526	526	556	556	556	579	579	579	554	554	554	526	526	526	523	523	523	603	603	603
Statistics:																														
Words Not Ignored:	1258	1258	1258	1190	1190	1190	1258	1258	1258	1286	1286	1286	1224	1224	1224	1166	1166	1166	1285	1285	1285	1262	1262	1262	1207	1207	1207	1287	1287	1287
Precision for POS Recognition (%):	22.3	22.2	22.6	41	41.3	43	24.2	24.2	25.4	56.8	57.4	56.5	54.3	54.6	55.4	56.9	57	56.5	21.6	21.6	20.2	62.1	61.6	61.1	63.7	64.1	61.9	24.2	24.3	23.2
Precision for Disambiguation (%):	10.2	10.2	15.3	18.2	18.2	28.7	9.3	9.38	16.8	24.8	24.6	39	23.6	24.1	39.1	24.6	24.1	39.1	7	7.32	13.7	26.5	26.5	40.7	28.1	28	41.4	9.95	9.87	14.8
Precision for Nouns (%):	72.6	72.6	78.6	62.3	62.7	75.5	53	53.7	64.5	62.3	62.1	71.8	59.9	59.7	69.9	60.1	60.3	70.8	54.9	54.1	63.9	71.3	70.7	82.8	75	74.8	82.8	58.9	58.9	63.6
Precision for Verbs (%):	66.7	65.5	73.9	91.4	92	78.5	90.6	89.6	84.9	87.9	84.7	73.8	90	89	75.7	87.7	87.5	76.3	94.7	88.1	72.2	87.6	89.3	82.4	88.4	84.3	75.6	84.7	85.6	71.2
Precision for Adjectives (%):	70.7	70.7	65.5	64.2	64.2	54.7	39.8	40.6	29.5	50.5	53.3	38.5	42.3	45.6	38.1	58.6	58.6	51.7	61.3	65.1	40.3	62	61.6	55.2	66.7	68.3	55.4	75.3	75.5	54.4
Precision for Adverbs (%):	66.7	66.7	66.7	77.4	80.6	77.4	68.7	62.5	54.2	68.6	69.8	62.8	76.8	76.8	63.8	79.7	79.7	73.4	71.9	75	46.9	72.9	72.9	67.5	69.9	71.8	66	68.6	68.6	60

	MIN	MAX	AVG
Words Not Ignored:			
Precision for POS Recognition (%):	20.2	64.1	42.8
Precision for Disambiguation (%):	7	41.4	21.8
Precision for Nouns (%):	53	82.8	66.1
Precision for Verbs (%):	67.3	92	84.2
Precision for Adjectives (%):	29.5	75.3	56.4
Precision for Adverbs (%):	46.9	80.6	69.4

FIGURE 2. Results