

Institut für Informatik der
Universität Zürich
Winterthurerstrasse 190
8057 Zürich

Fachgebiet Computerlinguistik
Prof. Dr. Michael Hess

Programmierprojekt
Erkennung von temporalen Ausdrücken

Bearbeiter: Stefan Höfler
Betreuer: Simon Clematide, Martin Volk

Schlussbericht über das Perl-Skript Tempex

Stefan Höfler
Wiesenbachstrasse 7a
9015 St. Gallen

shoefler@cl.unizh.ch

071 310 16 65

Inhalt

1	Einleitung	2
2	Material	2
3	Vorgehensweise von Tempex	2
3.1	Terminale und Lexika	2
3.2	Phrasen	2
3.3	Recall vs. Präzision	2
4	Adverbien	3
4.1	Adverbien-Lexikon	3
4.2	Temporalwert von Adverbien	3
4.3	Auswertung	3
4.4	Beobachtungen	4
4.4.1	Unterscheidung zwischen temporaler und modaler Bedeutung eines Adverbs	4
4.4.2	Unterscheidung zwischen temporaler und lokaler Bedeutung eines Adverbs	4
4.4.3	Adverb <i>wieder</i>	4
4.4.4	Adverb mit Negation	4
4.4.5	Adverb fälschlicherweise als Vergleichskonjunktion getaggt	4
4.4.6	Konjunktion fälschlicherweise als Adverb getaggt	4
4.5	Adverbialphrasen	5
4.5.1	Relevante Phrasenkomponenten	5
4.5.2	Unvollständig geparste Adverbialphrasen	5
5	Substantive und Nominalphrasen	6
5.1	Temporalwert von Substantiven und Nominalphrasen	6
5.2	Auswertung	6
5.3	Beobachtungen	7
5.3.1	Genitiv-Attribute	7
5.3.2	Kardinalzahlen	7
6	Präpositionen und Präpositionalphrasen	8
6.1	Präpositionen-Lexikon	8
6.2	Temporalwerte von Präpositionen und Präpositionalphrasen	8
6.3	Auswertung	8
6.4	Beobachtungen	9
6.4.1	Präpositionalphrasen mit Adverb	9
6.4.2	Zirkumposition rechts nicht Teil der PP	10
6.4.3	PP aufgrund der Länge nicht erkannt	10
6.4.4	Präpositionalphrasen mit mehr als einer Präposition	10
7	Literaturverzeichnis	11
8	Anhänge: Lexika und Programmcode	12

1 Einleitung

Das Perl-Skript Tempex kennzeichnet in einem im Negra-Format stehenden Satz potentiell temporale Ausdrücke.

Es wird der entsprechenden Komponente der Negra-Kommentar %%TEMP_n hinzugefügt, wobei n den Temporalwert kennzeichnet. Dabei gilt in Bezug auf die Temporalität eines Tokens (Wort oder Phrase): Je höher der Temporalwert %%TEMP_n, desto mehr Indizien dafür, dass das vorliegende Token temporal ist, sind vorhanden.

2 Material

Die beschriebenen Resultate basieren auf der Analyse von 200 Zeitungssätzen im Negra-Format. Die einzelnen Sätze durchliefen folgende Bearbeitungsschritte:

- Tagging der Wortkategorien (TreeTagger)
- NP/PP-Erkennung
- Konversion ins NEGRA Exportformat
- Erkennung temoraler Ausdrücke (Tempex)

3 Vorgehensweise von Tempex

3.1 Terminale und Lexika

Trifft Tempex eine Terminale beim token-weisen Durchgehen des Satzes an, so übergibt die Unteroutine `Annotate` das Wort und seine Wortart an die Unteroutine `LexCheck`. `LexCheck` überprüft das Wort auf sein Vorkommen im entsprechenden Lexikon.

Tempex verfügt über die folgenden Lexika:

- 1) Adverbien-Lexikon `temp_adv.lex`
- 2) Präpositionen-Lexikon `temp_praep.lex`
- 3) Substantiv-Lexikon `temp_noun.lex`

3.2 Phrasen

Phrasen werden von der Tempex-Unteroutine `Annotate` an die Unteroutine `PhraseCheck` weitergegeben. `PhraseCheck` übergibt die für die Anlyse relevanten Komponenten der Phrase an `LexCheck` und übernimmt die Summe der von den übergebenen Komponenten aus `LexCheck` zurückgelieferten Temporalwerten als eigenen Temporalwert.

3.3 Recall vs. Präzision

In den folgenden Auswertungen werden die Begriffe "Recall" und "Präzision" verwendet. Dabei ist der Präzisionswert auf 100 %, wenn sämtliche gefundenen Token auch gesucht sind, resp. auf 0 %, wenn keines der gefundenen Token gesucht ist.

Der Recallwert beträgt dagegen 100 %, wenn alle gesuchten Token auch gefunden wurden, resp. auf 0 %, wenn keines der gesuchten Token gefunden wurde.

4 Adverbien

4.1 Adverbien-Lexikon

Das Adverbien-Lexikon von Tempex umfasst folgende im Negra-Format unterschiedlich gekennzeichneten Wortarten:

- Adverbien: ADV
- adverbiale oder prädikative Adjektive: ADJD
- adverbiale Interrogativ- oder Relativpronomen: PWAV
- Pronominaladverbien: PAV

Im Folgenden werden diese Wortarten mit dem Überbegriff "Adverbien" bezeichnet.

4.2 Temporalwert von Adverbien

Adverbien sind im Adverbien-Lexikon `temp_adv.lex` mit den Werten 1 oder 2 versehen:

- 1: Das vorliegende Adverb kann temporal auftreten, könnte aber auch lokal oder modal verwendet sein, wie z.B. *da, dreimal*. (Die mögliche lokale Verwendung markiert Tempex mit einem zweiten Wert im Lexikon und dem Kommentar %%LOK_1.)
- 2: Das vorliegende Adverb kommt mit an Sicherheit grenzender Wahrscheinlichkeit temporal vor, wie z.B. *heute, jetzt*.

Dieses System hat zur Folge, dass Tempex in Bezug auf die Erkennung von Adverbien das Typs %%TEMP_1 stark übergeneriert, d.h. zwar einen guten Recall aber eine schlechte Präzision vorzuweisen hat.

4.3 Auswertung

Im analysierten Text sind 250 Token Adverbien. Davon versieht Tempex 48 mit dem Temporalwert 2 und 20 mit dem Temporalwert 1.

Sämtliche der mit Temporalwert 2 versehenen Token sind tatsächlich temporale Ausdrücke bzw. Teile von temporalen Ausdrücken. Es wurden keine eindeutig temporalen Adverbien nicht erkannt.

Die mit Temporalwert 1 versehenen Token fallen in den Grenzbereich zwischen modaler bzw. lokaler und temporaler Bedeutung:

da, häufig, noch, mal, mitunter, mehr, schon, weiter, wieder, zumal.

Die Einteilung dieser Adverbien ist eine Sache der definierenden Abgrenzung der möglichen Bedeutungen (siehe 4.4.1-2).

Tempex weist demnach für Adverbien mit Temporalwert 2 gute Präzision und guten Recall auf. Für Adverbien mit Temporalwert 1 ist der Recall zwar ebenfalls gut, die Präzision aber schlecht.

Tabelle 1: Auswertung Adverbien (Anzahl Token)

Text:	
Adverbien (ADV, ADJD, PWAV, PAV)	250
Analyse durch Tempex:	
Adverbien mit Temporalwert 2	48
richtig erkannt	48
falsch erkannt	0
nicht erkannte eindeutig temporal verwendete Adverbien	0

Präzision bei Adverbien mit Temporal Wert 2	100 %
Recall bei Adverbien mit Temporalwert 2	100 %
Adverbien mit Temporalwert 1	20
richtig erkannt	Angabe eines Wertes nicht möglich (siehe 4.4)
falsch erkannt	Angabe eines Wertes nicht möglich (siehe 4.4)
nicht erkannte eindeutig temporal verwendete Adverbien	0
Präzision bei Adverbien mit Temporal Wert 1	keine Angaben
Recall bei Adverbien mit Temporalwert 1	100 %

4.4 Beobachtungen

4.4.1 Unterscheidung zwischen temporaler und modaler Bedeutung eines Adverbs

Bei einigen Adverbien ist die Unterscheidung zwischen modaler und temporaler Bedeutung nicht immer klar. Dabei können zwei Fälle unterschieden werden:

- 1) Die Einteilung des Adverbs ist allgemein unklar. Es fehlt eine Definition.
- 2) Die Bedeutung des Adverbs ist im konkreten Fall klar, aber es fehlt eine (syntaktische) Regel, mit der diese Bedeutung "errechnet" werden kann.

Beispiele: *häufig, noch, mal, mitunter, mehr, schon, wieder, zumal*

4.4.2 Unterscheidung zwischen temporaler und lokaler Bedeutung eines Adverbs

Für Adverbien, die sowohl temporale als auch lokale Bedeutung tragen können, fehlt eine (syntaktische) Unterscheidungsregel (vgl. 4.4.1). Allenfalls können Stellungsregularitäten berücksichtigt werden: Lokale Adverbialen stehen immer nach temporalen Adverbialen. Aber wie kann eine Unterscheidung erfolgen, wenn nur eine Adverbiale vorhanden ist?

Beispiele: *da, weiter*

4.4.3 Adverb wieder

Das Adverb *wieder* kann oft nicht schlüssig entweder als selbständiges Adverb oder als Verbzusatz interpretiert werden. Das ist allerdings ein Problem des Taggers.

4.4.4 Adverb mit Negation

Soll die Negation eines temporalen Adverbs diesem zugeordnet werden? Dies kann der Benutzer von Tempex mit Hilfe der bei Start des Programms erfragten Programm-Option selber beeinflussen. Wird die Negation zum Adverb gezählt, besteht die Gefahr, dass auch Negationen, die nur zufällig vor einem Adverb stehen, sich aber nicht auf dieses beziehen, zu einer Phrase mit dem folgenden Adverb zusammen gefasst werden.

Dies ist ein Problem der Phrasen-Erkennung, siehe auch 4.5.2 b).

4.4.5 Adverb fälschlicherweise als Vergleichskonjunktion getaggt

Wenn der verwendete Tagger Adverbien fälschlicherweise als Vergleichskonjunktionen (KOKON) analysiert, entstehen unter Umständen auch Schwierigkeiten bei der Phrasen-Erkennung:

Beispiel: *wie lange* --> KOKON, ADV anstatt AVP(ADV, ADV).

4.4.6 Konjunktion fälschlicherweise als Adverb getaggt

Fälschlicherweise als Adverb getaggte Konjunktionen können zu falschen Phrasen-Konstrukten führen. Das ist ein Problem der Phrasen-Erkennung und des Taggers.

Beispiel:

(1)	#BOS 5132 0 900000000 1					
	Da	ADV	--	MO	500	%%TEMP_1
	künftig	ADJD	--	HD	500	%%TEMP_2
	das	ART	--	NK	504	
	Internet	NN	--	NK	504	
	als	KOKOM	--	--	0	
	Netz	NN	--	--	0	
	für	APPR	--	AC	503	%%TEMP_1
	die	ART	--	NK	503	
	weltweite	ADJA	--	NK	503	
	Transaktions-	TRUNC	--	CJ	502	
	und	KON	--	CD	502	
	Message-Verarbeitung	NN	--	CJ	502	
	dienen	VVINF	--	--	0	
	werde	VAFIN	--	--	0	
	,	\$,	--	--	0	
	...					
	#500	AP	--	--	0	%%TEMP_2
	...					

4.5 Adverbialphrasen

4.5.1 Relevante Phrasenkomponenten

Die für die Analyse der Temporalität relevanten Komponenten einer Adverbialphrase sind diejenigen mit den Kantenbezeichnungen "adverbial phrase component" (AVC) oder "head" (HD).

4.5.2 Unvollständig geparste Adverbialphrasen

Weil das vorliegende Textmaterial keine Adverbialphrasen enthält (lediglich eine NP/PP-Erkennung wurde durchgeführt), ergänzt Tempex diese Struktur: Beim Auftreten eines temporalen Adverbs wird das vorangehende Wort durch die aus der Unteroutine Annotate heraus aufgerufene Unteroutine PrevCheck geprüft.

a) Adverbialphrasen

Handelt es sich beim vorangehenden Wort um ein Adverb, wird es an LexCheck zur Überprüfung im Lexikon `prev_adv.lex` (previous adverbs lexicon) übergeben. In diesem Lexikon sind modifizierende Adverbien wie z.B. *sehr*, *so*, *zu* etc. erfasst. Ist die Überprüfung positiv, erstellt Tempex einen neuen Parent-Knoten (mit fortlaufender 600er-Nummer) für eine neue Adverbialphrase folgenden Musters:

AVP -> ADV(MO), ADV(HD).

b) Negierte Adverbien

Handelt es sich beim vorangehenden Wort um die Negation *nicht* (PTKNEG), erstellt Tempex - falls die entsprechende Programm-Option aktiviert ist (siehe 4.4.4) - einen neuen Parent-Knoten (mit fortlaufender 600er-Nummer) für eine neue Adverbialphrase folgenden Musters:

AVP -> PTKNEG(MO), ADV(HD).

c) Adjektivphrasen

Im Negra-Format können "Adverbien" vom Typ ADJD (adverbial oder prädikativ verwendete Adjektive) zusammen mit einem Adverb (ADV) eine Adjektivphrase (AP) bilden.

Tempex ergänzt für diese Konstruktion analog zu a) eine Adverbialphrase folgenden Musters:

AVP -> ADV(MO), ADJD(HD).

Beispiel: *sehr schnell*

5 Substantive und Nominalphrasen

5.1 Temporalwert von Substantiven und Nominalphrasen

Da jedes Substantiv, das in einem temporalen Ausdruck vorkommen kann, grundsätzlich auch ausserhalb eines solchen auftreten kann, sind alle potentiell temporalen Substantive im Lexikon `temp_noun.lex` lediglich mit dem Temporalwert 1 versehen.

Beispiel:

Diese Nacht (temporal) *kommt er bestimmt.* vs. *Diese Nacht* (nicht temporal) *ist schwarz.*

Nominalphrasen erhielten demnach erst dann einen Temporalwert, der höher als 1 wäre, wenn z. B. zusätzlich auch Adjektive analysiert würden, was aber in der vorliegenden Version von Tempex nicht der Fall ist.

5.2 Auswertung

Von den im analysierten Text vorhandenen insgesamt 1063 Token, die als Substantive (NN) markiert sind, hat Tempex 47 mit dem Temporalwert 1 gekennzeichnet.

Davon sind 32 Token tatsächlich Teil eines temporalen Ausdrucks, 31 Token als Teil einer Präpositionalphrase (siehe Abschnitt 6) und 1 Token als Teil einer Nominalphrase (NP):

(2)

#BOS 5058 0 900000000 1						
Die	ART	--	--	0		
ersten	ADJA	--	--	0		
beiden	PIDAT	--	NK	501		
numerischen	ADJA	--	NK	501		
Consumergeräte	NN	--	NK	501		
kommen	VVFIN	--	--	0		
dieser	PDAT	--	NK	502		
Tage	NN	--	NK	502	%%TEMP_1	
in	APPR	--	AC	500	%%TEMP_1	%%LOK_1
die	ART	--	NK	500		
Geschäfte	NN	--	NK	500		
.	\$.	--	--	0		
#500	PP	--	--	0	%%TEMP_1	%%LOK_1
#501	NP	--	--	0		
#502	NP	--	--	0	%%TEMP_1	
#EOS 5058						

Nicht erkannt wurden die in temporalen Präpositionalphrasen vorkommenden Substantive *ISO-900-Re-Audit* und *Börseneinführung* (vgl. 6.3).

Von 476 im Text vorkommenden Nominalphrasen versieht Tempex 14 mit dem Temporalwert 1 und 0 mit dem Temporalwert 2. Davon ist lediglich die in Beispiel (2) gezeigte Nominalphrase mit Temporalwert 1 auch tatsächlich ein temporaler Ausdruck.

Tabelle 2: Auswertung Substantive (Anzahl Token)

Text:	
Substantive (NN)	1'063
Substantive in temporalen Ausdrücken	34
Analyse durch Tempex:	
als temporal analysierte Substantive (Wert 1)	47
richtig erkannt	32
falsch erkannt	15
nicht erkannte temporale Substantive	2
Präzision bei Substantiven mit Temporalwert 1	68 %
Recall bei Substantiven mit Temporalwert 1	94 %

Tabelle 3: Auswertung Nominalphrasen (Anzahl Token)

Text:	
Nominalphrasen (NP)	476
Temporale Nominalphrasen	1
Analyse durch Tempex:	
als temporal analysierte Nominalphrasen (Wert 1)	15
richtig erkannt	1
falsch erkannt	14
nicht erkannte temporale Nominalphrasen	0
Präzision bei Nominalphrasen mit Temporalwert 1	6.7 %
Recall bei Nominalphrasen mit Temporalwert 1	100 %

5.2 Beobachtungen

5.3.1 Genitiv-Attribute

Genitiv-Attribute werden vom NP/PP-Tool nicht erkannt. Wäre dies der Fall, würde Tempex den folgenden Nominalphrasen mit Genitiv-Attribut den Temporalwert 2 zuweisen:

(3)

#BOS 5144 0 900000000 1						
Das	ART	--	NK	501		
niederländische	ADJA	--	NK	501		
Electronic-Money-Unternehmen	NN	--	NK	501		
Digicash	NE	--	--	0		
will	VMFIN	--	--	0		
Anfang	NN	--	--	0	%%TEMP_1	
des	ART	--	NK	502		
nächsten	ADJA	--	NK	502		
Jahres	NN	--	NK	502	%%TEMP_1	
E-Cash	NE	--	--	0		
in	APPR	--	AC	500	%%TEMP_1	%%LOK_1
Japan	NE	--	NK	500		
ausgeben	VVINFIN	--	--	0		
.	\$.	--	--	0		
#500	PP	--	--	0	%%TEMP_1	%%LOK_1
#501	NP	--	--	0		
#502	NP	--	--	0	%%TEMP_1	
#EOS 5144						

(4)

#BOS 5161 0 900000000 1						
...						
Festnetzlizenzen	NN	--	NK	501		
,	\$.	--	--	0		
die	PRELS	--	--	0		
im	APPRART	--	AC	503	%%TEMP_1	%%LOK_1
Juli	NN	--	NK	503	%%TEMP_1	
dieses	PDAT	--	NK	506		
Jahres	NN	--	NK	506	%%TEMP_1	
ausgeschrieben	VVPP	--	--	0		
wurden	VAFIN	--	--	0		
,	\$.	--	--	0		
...						
#503	PP	--	--	0	%%TEMP_2	%%LOK_1
#504	PP	--	--	0	%%TEMP_2	%%LOK_1
#505	NP	--	--	0		
#506	NP	--	--	0	%%TEMP_1	
#EOS 5161						

5.3.2 Kardinalzahlen

Tempex weist vierstelligen Kardinalzahlen (CARD) vom Typ 19.. und 20.. den Temporalwert 2 zu, Kardinalzahlen vom Typ 1... erhalten den Temporalwert 1. Damit werden die gängigsten Jahreszahlen erkannt.

Problematisch ist allerdings, dass der NP/PP-Erkennen Kardinalzahlen nicht als Teil der dazugehörigen NP resp. PP analysiert:

```
(5) #BOS 5035 0 9000000000 1
      Typo          NE          -- --          0
      Art           NN          -- --          0
      arbeitete     VVFIN       -- --          0
      bis           APPR        -- AC          504      %%TEMP_1      %%LOK_1
      Anfang        NN          -- NK          504      %%TEMP_1
      1996          CARD        -- --          0          %%TEMP_2
      mit           APPR        -- AC          503      %%TEMP_1
      Macintosh-Computern  NN    --          CJ          500
      ...
      #504          PP          -- --          0          %%TEMP_2      %%LOK_1
      ...
      #EOS 5035
```

6 Präpositionen und Präpositionalphrasen

6.1 Präpositionen-Lexikon

Das Präpositionen-Lexikon von Tempex umfasst folgende im Negra-Format unterschiedlich markierte Wortarten:

- Präpositionen, Zirkumpositionen links: APPR
- Postpositionen: APPO
- Zirkumpositionen rechts: APZR
- Präpositionen mit Artikel: APPRART

Im Folgenden werden diese Wortarten mit dem Überbegriff "Präpositionen" bezeichnet.

6.2 Temporalwerte von Präpositionen und Präpositionalphrasen

Präpositionen sind im Lexikon temp_praep.lex mit den Werten 1 und 2 versehen:

- 1: Die Präposition *kann* temporale Bedeutung haben, könnte aber auch z.B. lokal oder modal verwendet sein, wie z.B. *in*, *auf*, *mit*.. (Die mögliche lokale Verwendung markiert Tempex mit einem zweiten Wert im Lexikon und dem Kommentar %%LOK_1.)
- 2: Die vorliegende Präposition ist mit an Sicherheit grenzender Wahrscheinlichkeit temporal verwendet, wie z.B. *innen*, *während*, *zeit*.

Der Temporalwert einer Präpositionalphrase (PP) ist die Summe der Temporalwerte ihrer für die Temporalbestimmung relevanten Komponenten, d.h. im allgemeinen die Summe des Temporalwertes der Präposition (Komponente mit der Kante "adpositional case marker" AC) und jenes ihres Substantives ("noun kernel modifier" NK) - oder Adverbs (head "HD").

6.3 Auswertung

Von den insgesamt 514 im analysierten Text vorkommenden Token, die als Präpositionen markiert sind, teilt Tempex 490 Token den Temporalwert 1 und zwei Token den Temporalwert 2 zu. Dabei sind 33 der 490 Token mit Temporalwert 1 tatsächlich temporal verwendete Präpositionen, ebenso wie die 2 Token mit Temporalwert 2.

Tabelle 4: Auswertung Präpositionen (Anzahl Token)

Text:	
Präpositionen (APPR, APPO, APZR, APPRART)	514
Präpositionen in temporalen Ausdrücken	35
Analyse durch Tempex:	
als temporal analysierte Präpositionen (Wert 1)	490
richtig erkannt	35
falsch erkannt	455
nicht erkannte temporale Präpositionen	0
Präzision bei Präpositionen mit Temporalwert 1	7 %
Recall bei Präpositionen mit Temporalwert 1	100 %

Der analysierte Text enthält insgesamt 469 als Präpositionalphrase (PP) markierte Token. 417 dieser Token erhalten von Tempex den Temporalwert 1 zugewiesen, wovon lediglich 2 auch wirklich temporale Ausdrücke sind. 31 Token sind mit dem Temporalwert 2 markiert, davon sind 28 tatsächlich temporale Ausdrücke.

Fünf temporale Präpositionalphrasen konnten von Tempex nicht analysiert werden, weil sie der NP/PP-Erkenner aufgrund ihrer Länge nicht als PP markiert hat (siehe 6.4.3). Drei Präpositionalphrasen wurden aufgrund der enthaltenen Substantive resp. Substantivteile mit dem Temporalwert 2 versehen:

von desssen gesamter Arbeitszeit (Satz 5038), *in Metaphase* (Satz 5098), *bei einer beachtlichen Gewinnspanne* (Satz 5187).

Tabelle 5: Auswertung Präpositionalphrasen (Anzahl Token)

Text:	
Präpositionalphrasen (PP)	469
temporale Präpositionalphrasen	30
Analyse durch Tempex:	
als temporal analysierte PP (Wert 2)	31
richtig erkannt	28
falsch erkannt	3
nicht erkannte temporale PP	2
Präzision bei PP mit Temporalwert 2	90 %
Recall bei PP mit Temporalwert 2	93 %
als temporal analysierte PP (Wert 1)	417
richtig erkannt	2
falsch erkannt	415
nicht erkannte temporale PP	0
Präzision bei PP mit Temporalwert 1 oder 2	0.5 %
Recall bei PP mit Temporalwert 1 oder 2	100 %

6.4 Beobachtungen

6.4.1 Präpositionalphrasen mit Adverb

Der NP/PP-Erkenner "verpasst" Präpositionalphrasen mit Adverbien. Tempex ergänzt ein temporales Adverb, vor dem eine "lose" Präposition steht, zu einer Präpositionalphrase (mit fortlaufender 600er-Nummer) folgenden Musters:

PP-> APPR(AC), ADV(HD).

6.4.2 Zirkumposition rechts nicht Teil der PP

Zirkumpositionen rechts (APZR) werden vom NP/PP-Erkenner nicht der PP zugeordnet.

Beispiel:

(6)

#BOS 5102 0 900000000 1						
Von	APPR	--	AC	500	%%TEMP_1	%%LOK_1
Beginn	NN	--	NK	500	%%TEMP_1	
an	APZR	--	--	0	%%TEMP_1	%%LOK_1
mischte	VVFIN	--	--	0		
CSC	NE	--	--	0		
Ploenzke	NN	--	--	0		
im	APPRART	--	AC	501	%%TEMP_1	%%LOK_1
EDM-Segment	NN	--	NK	501		
als	KOKOM	--	--	0		
Berater	NN	--	--	0		
mit	PTKVZ	--	--	0		
.	\$.	--	--	0		
#500	PP	--	--	0	%%TEMP_2	%%LOK_1
#501	PP	--	--	0	%%TEMP_1	%%LOK_1
#EOS 5102						

6.4.3 PP aufgrund der Länge nicht erkannt

Das NP/PP-Tool erkennt Präpositionalphrasen mit längeren nominalen Bestandteilen nicht.

Beispiel:

(7)

#BOS 5068 0 900000000 1						
...						
in	APPR	--	--	0	%%TEMP_1	%%LOK_1
den	ART	--	--	0		
letzten	ADJA	--	--	0		
vier	CARD	--	NK	505		
Jahren	NN	--	NK	505	%%TEMP_1	
...						
#505	NP	--	--	0	%%TEMP_1	
#EOS 5068						

6.4.4 Präpositionalphrasen mit mehr als einer Präposition

Wenn vor einer Präpositionalphrase ein "lose" Präposition steht, wird diese von Tempex zur PP hinzugenommen. PP mit mehr als einer Präposition werden vom NP/PP-Erkenner nicht vollständig erkannt.

Beispiel:

(8)

#BOS 5187 0 900000000 1						
Bis	APPR	--	--	0	%%TEMP_1	%%LOK_1
zum	APPRART	--	AC	502	%%TEMP_1	%%LOK_1
Jahr	NN	--	NK	502	%%TEMP_1	
2000	CARD	--	--	0	%%TEMP_1	
...						
#502	PP	--	--	0	%%TEMP_2	%%LOK_1
...						

7 Literaturverzeichnis

Dreyer, Hilke und Schmitt, Richard, *Lehr- und Übungsbuch der deutschen Grammatik*, Verlag für Deutsch, München, 1991.

Drosdowski, Günter et. al. (Eds.), *Duden Grammatik der deutschen Gegenwartssprache*, Duden Bd. 4, Dudenverlag, Mannheim, Wien, Zürich, 1984.

Schröder, Jochen, *Lexikon der deutschen Präpositionen*, Verlag Enzyklopädie, Leipzig, 1990.

Anhang A: Lexika

Das Adverbien-Lexikon: temp_adv.lex

In der ersten Spalte nach dem Adverb ist der Temporalwert eingetragen, in der zweiten Spalte ein allfälliger Lokalwert (siehe 4.2)

zwischenzeitlich	1	0	donnerstags	2	0
ab und zu	2	0	dreimal	2	0
abend	2	0	durchweg	1	0
abends	2	0	dutzendmal	2	0
abermals	2	0	eben	1	0
achtmal	2	0	ebenda	2	0
achtmillionenmal	2	0	ehedem	2	0
achtzigmal	2	0	ehegestern	2	0
allabends	2	0	ehemals	2	0
alda	1	1	eher	1	0
alldieweil	1	0	ein andermal	2	0
allemaal	1	0	einigemal	2	0
allerfrühestens	2	0	einmal	1	0
allermeist	2	0	einst	2	0
allerspättest	2	0	einstmals	2	0
allerspättestens	2	0	einstweilen	2	0
allerweil	1	0	endlich	2	0
alleweil	1	0	erst	2	0
allezeit	2	0	erstmals	2	0
alltags	2	0	etlichemal	2	0
allweil	1	0	ewig	2	0
allzeit	2	0	ewiglich	2	0
allzuoft	2	0	feiertags	2	0
allzuviel	1	0	fortab	2	0
anderentags	2	0	fortan	2	0
anfangs	2	0	forthin	2	0
angelegentlich	2	0	freitags	2	0
anhin	1	0	früh	2	0
anno	2	0	frühauf	2	0
ansonst	1	0	früher	2	0
ansonsten	1	0	frühestens	2	0
auf einmal	1	0	frühmorgens	2	0
augenblicklich	2	0	ganztags	2	0
augenblicks	2	0	gerade	1	0
bald	2	0	gestern	2	0
baldigst	2	0	gleich	1	0
beizeiten	2	0	gleichzeitig	2	0
bereits	2	0	hernach	2	0
bisher	2	0	heuer	2	0
bislang	2	0	heute	2	0
bisweilen	2	0	heutigentags	2	0
da	1	0	heutzutage	2	0
damals	2	0	hierauf	2	0
danach	2	0	hiervor	1	1
dann	2	0	hierzwischen	1	1
dann und wann	2	0	hin und wieder	2	0
darauf	2	0	hinterdrein	1	1
daraufhin	2	0	hinterher	1	1
das erstmal	2	0	hinweg	1	1
davor	1	0	hundertmal	2	0
dazwischen	1	0	immer	2	0
demnächst	2	0	immerfort	2	0
dereinst	2	0	immerzu	2	0
dermaleinst	2	0	indes	1	0
derweil	2	0	indessen	1	0
derweilen	2	0	inzwischen	2	0
derzeit	2	0	irgendeinmal	2	0
dienstags	2	0	irgendwann	2	0
diesmal	2	0	jahraus	2	0

jahrein	2	0	öfters	2	0
je	1	0	oftmals	2	0
jederzeit	2	0	paarmal	2	0
jederzeitig	2	0	quartalsweise	1	0
jedesmal	2	0	rechtzeitig	2	0
jeher	2	0	saisonweise	1	0
jemals	2	0	samstags	2	0
jetzt	2	0	schliesslich	1	0
jeweils	2	0	schlussendlich	1	0
jüngst	2	0	schnell	2	0
jüngstens	2	0	schnellstens	2	0
jüngsthin	2	0	schnurstracks	2	0
just	1	0	schon	1	0
justament	1	0	seinerzeit	2	0
kaum	1	0	seitdem	2	0
keinmal	2	0	seither	2	0
künftig	2	0	selten	2	0
kurzhin	1	0	sodann	2	0
kürzlich	2	0	soeben	2	0
lange	2	0	sofort	2	0
langhin	1	0	sogleich	2	0
längst	2	0	sommers	2	0
längstens	2	0	sommersüber	2	0
letztamal	2	0	sommertags	2	0
letztendlich	1	0	sonnabends	2	0
letztens	2	0	sonntags	2	0
letztthin	2	0	sonntags	2	0
letztlich	1	0	sonst	1	0
letztmals	2	0	spät	2	0
mal	1	0	spätabends	2	0
manchmal	2	0	später	2	0
mehr	1	0	späterhin	2	0
mehrmals	2	0	spätestens	2	0
meist	2	0	spätnachmittags	2	0
meistens	2	0	stets	2	0
millionenmal	2	0	stundenweise	2	0
mittag	2	0	tagaus	2	0
mittags	2	0	tagein	2	0
mitternachts	2	0	tageweise	2	0
mittlerweile	2	0	tags	2	0
mittsommers	2	0	tagsüber	2	0
mittwochs	2	0	übermorgen	2	0
mitunter	1	0	überübermorgen	2	0
monatsweise	1	0	ultimo	1	0
montags		20	unendlichemal	2	0
morgen	2	0	unendlichmal	2	0
morgens	2	0	unlängst	2	0
nachgerade	1	0	unterdes	2	0
nachher	2	0	unterdessen	2	0
nachhinein	1	1	untertags	2	0
nachmittags	2	0	unterweilen	2	0
nächstens	2	0	unversehens	1	0
nachts	2	0	unzähligemal	2	0
nachtsüber	2	0	verschiedentlich	1	0
nebenher	1	1	vielmals	1	0
neuerdings	2	0	viermal	1	0
neuestens	2	0	vorab	2	0
neulich	2	0	voraus	1	1
neustens	2	0	vorbei	1	1
nie	2	0	vordem	2	0
niemals	2	0	vorerst	1	0
nimmer	2	0	vorgängig	2	0
nimmermehr	2	0	vorgestern	2	0
nirgendwann	2	0	vorher	2	0
noch	1	0	vorhin	2	0
nochmals	2	0	vormals	2	0
nun	2	0	vormittag	2	0
nunmehr	2	0	vormittags	2	0
oft	2	0	vorneweg	2	0
öfter	2	0	vornherein	1	0

vornhin	1	1	zuallerletzt	2	0
vornweg	1	1	zuallermeist	1	0
vorüber	1	1	zuerst	2	0
vorvorgestern	2	0	zugleich	2	0
vorweg	1	1	zukünftig	2	0
vorzeiten	2	0	zuletzt	2	0
währenddem	2	0	zumeist	1	0
währenddes	2	0	zunächst	1	0
währenddessen	2	0	zurzeit	2	0
wann	2	0	zuvor	2	0
weiter	1	1	zuweilen	2	0
weiterhin	2	0	zweimal	1	0
werktags	2	0	zwischen	1	1
wie lange	2	0	zwischenzeitlich	2	0
wie oft	2	0	täglich	2	0
wieder	1	0	wöchentlich	2	0
wiederum	2	0	monatlich	2	0
wievielmal	2	0	stündlich	2	0
winters	2	0	jährlich	2	0
wintersüber	2	0	pünktlich	2	0
wochentags	2	0	häufig	1	0
wochenweise	2	0	länger	1	1
wonach	1	1	lang	2	0
woraufhin	2	0	regelmässig	1	0
wozwischen	1	1	ständig	2	0
zeitlebens	2	0	nachträglich	2	0
zeitweise	2	0	allmählich	1	0
zigmal	1	0	dauernd	2	0
zuallerallerletzt	2	0			

Das Substantive-Lexikon: temp_noun.lex

Die erste Spalte nach dem Substantiv gibt den Temporalwert an. Dieser beträgt immer 1, denn kein Substantiv kann ausschliesslich in temporaler Verwendung vorkommen. Die Werte für den Lokalwert (2. Spalte) sind für Substantive nicht erfasst.

Für die Erkennung eines Substantives ist nur der letzte Teil des Wortes relevant: D.h. mit dem Eintrag *zeit* erkennt Tempex sowohl *Zeit* als auch *Jahreszeit*, *Tageszeit*, *Schulzeit*, *Regenzeit*, etc.

quartal	1	0	monat	1	0
quartale	1	0	monats	1	0
quartalen	1	0	monate	1	0
quartals	1	0	monaten	1	0
beginn	1	0	januar	1	0
beginns	1	0	januars	1	0
anfang	1	0	februar	1	0
anfänge	1	0	februars	1	0
anfängen	1	0	märz	1	0
anfangs	1	0	april	1	0
mittwoch	1	0	aprils	1	0
mitwoche	1	0	juni	1	0
mittwochs	1	0	junis	1	0
mittwochen	1	0	juli	1	0
stunde	1	0	julis	1	0
stunden	1	0	august	1	0
minute	1	0	augusts	1	0
minuten	1	0	september	1	0
sekunde	1	0	septembers	1	0
sekunden	1	0	oktober	1	0
tag	1	0	oktobers	1	0
tages	1	0	november	1	0
tage	1	0	novembers	1	0
tagen	1	0	dezember	1	0
woche	1	0	dezembers	1	0
wochen	1	0	jahr	1	0
ende	1	0	jahres	1	0
endes	1	0	jahre	1	0
enden	1	0	jahrs	1	0

jahren	1	0	phase	1	0
frühling	1	0	phasen	1	0
frühlings	1	0	äon	1	0
sommer	1	0	äons	1	0
sommers	1	0	äonen	1	0
herbst	1	0	milleniums	1	0
herbsts	1	0	millenium	1	0
herbstes	1	0	säkulum	1	0
winter	1	0	säkulum	1	0
winters	1	0	dezennium	1	0
tags	1	0	dezenniums	1	0
jahrzehnt	1	0	semester	1	0
jahrzehnts	1	0	semesters	1	0
jahrzehnte	1	0	trimester	1	0
jahrzehnten	1	0	trimesters	1	0
jahrhundert	1	0	zeitabstand	1	0
jahrhunderts	1	0	zeitabstands	1	0
jahrhunderte	1	0	zeitabstände	1	0
jahrhunderten	1	0	zeitabständen	1	0
jahrtausend	1	0	vergangenheit	1	0
jahrtausends	1	0	vergangenheiten	1	0
jahrtausende	1	0	intervall	1	0
jahrtausenden	1	0	intervalle	1	0
weile	1	0	intervallen	1	0
moment	1	0	intervalls	1	0
moments	1	0	zeitraum	1	0
momente	1	0	zeitraums	1	0
momenten	1	0	zeiträume	1	0
augenblick	1	0	zeiträumen	1	0
augenblicks	1	0	gegenwart	1	0
augenblicke	1	0	heute	1	0
augenblicken	1	0	jetzt	1	0
zeit	1	0	moderne	1	0
zeiten	1	0	postmoderne	1	0
uhr	1	0	zukunft	1	0
epoche	1	0	dauer	1	0
epochen	1	0	spanne	1	0
alter	1	0	spannen	1	0
alters	1	0	silvester	1	0
altern	1	0	silvesters	1	0
ewigkeit	1	0	weihnacht	1	0
ewigkeiten	1	0	weihnachten	1	0
zeitpunkt	1	0	ostern	1	0
zeitpunkts	1	0	pfingsten	1	0
zeitpunkte	1	0	abend	1	0
zeitpunkten	1	0	abende	1	0
frist	1	0	abenden	1	0
fristen	1	0	abends	1	0
termin	1	0	gestern	1	0
termins	1	0	morgen	1	0
termine	1	0	morgens	1	0
terminen	1	0	mittag	1	0
ära	1	0	mittags	1	0
zeitabschnitt	1	0	mittage	1	0
zeitabschnitts	1	0	mittagen	1	0
zeitabschnitte	1	0	nacht	1	0
zeitabschnitten	1	0	nachts	1	0
periode	1	0	nächte	1	0
perioden	1	0	nächten	1	0

Das Präpositionen-Lexikon: temp_praep.lex

In der ersten Spalte nach der Präposition ist der Temporalwert eingetragen. Bei Temporalwert 2 hat die Präposition ausschliesslich temporale Bedeutung. Die zweite Spalte gibt einen allfälligen Lokalwert an, falls die Präposition auch in lokaler Bedeutung vorkommen kann (siehe [6.2](#)).

ab	1	1	am	1	1
----	---	---	----	---	---

an	1	1	mit	1	0
auf	1	1	nach	1	1
aus	1	1	seit	1	0
ausserhalb	1	1	über	1	1
bei	1	1	um	1	1
beim	1	1	unter	1	1
binnen	2	0	von	1	1
bis	1	1	vor	1	1
durch	1	1	während	2	0
für	1	0	zeit	2	0
hindurch	1	1	zu	1	1
gegen	1	1	zum	1	1
im	1	1	zur	1	1
in	1	1	zwischen	1	1
innerhalb	1	1			

Lexikon für modifizierende Adverbien: prev_adv.lex

Dieses Lexikon wird von der Unterroutiner PrevCheck zur Ergänzung von unvollständig geparsten Adverbial- oder Präpositionalphrasen verwendet. Steht vor einer solchen Phrase ein Adverb aus prev_adv.lex, wird es mit der Phrase zusammengefasst (siehe auch [4.5.2](#)).

beinahe	1	1	schon	1	1
bereits	1	1	sehr	1	1
erst	1	1	so	1	1
etwa	1	1	spätestens	1	1
fast	1	1	wenigstens	1	1
frühestens	1	1	ziemlich	1	1
höchstens	1	1	zirka	1	1
midestens	1	1			
nahezu	1	1			

```

#####
# Tempex Version 6.2
# Perl-Skript zur Markierung temporaler Ausdrücke in Sätzen im Negra-Format
#####
# Die von Tempex erwartete Verzeichnisstruktur:
# tempex
#      |-bin           Programmcode
#      |-input        der zu analysierende Negra-Input
#      |-lex          Lexikon-Dateien
#      |-log          Protokoll-Dateien (wird von Tempex erstellt)
#      |-output       der Programm-Output (wird von Tempex erstellt)
#####
# Benennung der Tempex-Variablen für das Negra-Format:
# Beispiel:
#-----
# #BOS...
# ...
# in          APPR  --      AC      504
# dem        ART   --      NK      504
# Jahr       NN    --      NK      504
# ...
# #504       PP    --      --      0
# #EOS...
#-----
# ganzer Satz: globale Variable @TheSentence
# Zeile: $TheSentence[i] bzw. $Line bzw. @Token
# 1.Spalte: $Node bzw. $Token[0]
# 2.Spalte: $Pos bzw. $Token[1]
# 3.Spalte: $Token[2]
# 4.Spalte: $Edge bzw. $Token[3]
# 5.Spalte: $Parent bzw. $Token [4]
#####

#!/usr/bin/perl

package Tempex;
use strict 'vars';

#-----#
# Erfragen der Programm-Optionen
#-----#

#Variablen-Deklaration
my $NegOption; #Programm-Option (Werte 'y' oder 'n')

print "Programm-Option:\nSoll Tempex \"nicht!\"+Adverb zu einer Adverbialphrase
zusammenfassen? (y/n)\n";
$NegOption=<STDIN>;
chomp $NegOption;
until(($NegOption eq 'y')or($NegOption eq 'n')) {
    print "Bitte geben sie entweder y oder n ein:\n";
    $NegOption=<STDIN>;
    chomp $NegOption;
}

#-----#
# Definition der verwendeten Dateien und Pfade
#-----#

# Definition der Input-Datei
my $Negra="..\input\zeitung.txt";
print "\nTempex 6.2 bearbeitet die Datei $Negra\n";

# Definition der Lexikon-Dateien
my $LexDir="..\lex"; #Pfadangabe für Lexikondateien
my $AdvLex="$LexDir\temp_adv.lex"; #Lexikon: ADV, ADJD, PWAV, PAV
my $NnLex="$LexDir\temp_noun.lex"; #Lexikon: NN
my $PrevAdvLex="$LexDir\prev_adv.lex"; #Lexikon mit Gradadverbien
my $PraepLex="$LexDir\temp_praep.lex"; #Lexikon: APPR, APPO, APZR, APPRART

# Konvertierung der Lex-Dateien in Lex-Hashes mittels Unterroutine LexToHash
# (mit der Lexikon-Datei und einer festen Referenz auf den Lexikon-Hash)
my %AdvHash=();
my %NnHash=();
my %PrevAdvHash=();
my %PraepHash=();
&LexToHash($AdvLex, \%AdvHash);
&LexToHash($NnLex, \%NnHash);
&LexToHash($PrevAdvLex, \%PrevAdvHash);
&LexToHash($PraepLex, \%PraepHash);

#Definition der Datei für den Programm-Output
my $OutputDir="..\output";
mkdir("$OutputDir", 0770);
my $Output="$OutputDir\result.txt";

#Definition der Protokoll-Dateien
my $LogDir="..\log";

```

```

mkdir("$LogDir", 0770);
my $Log="$LogDir\protok.log";

# Oeffnen der Dateien
open(LOG, ">$Log") or die "Log-File konnte nicht erstellt werden\n";
open(NEGRA, "$Negra") or die "Datei $Negra nicht gefunden\n";
open(OUT, ">$Output") or die "Datei $Output konnte nicht erstellt werden\n";

#-----#
# Hauptroutine: satzweise Überprüfung auf temporale Ausdrücke
#-----#

#Variablen-Deklaration
@Tempex::TheSentence=();          #(globaler) Array zur Speicherung des Satzes

while(<NEGRA>) {                  #bearbeitet die Input-Datei zeilenweise
    #zeilenweises Einlesen des Satzes
    my $TheLine=$_;              #speichert die aktuelle Zeile
    chomp($TheLine);             #entfernt die Zeilenschaltung
    push(@Tempex::TheSentence, $TheLine); #fügt die Zeile(=Token)
                                     #hinten/unten an den
                                     #Satz-Array an

    #Bearbeitung des Satzes
    if($TheLine=~m/#EOS/) {      #wenn ein Satzende(EOS-Zeile) erreicht
        &Annotate;               #Aufruf der Unterroutine Annotate
        #Ausgabe des annotierten Satzes:
        print OUT join("\n", @Tempex::TheSentence); print OUT "\n";
        @Tempex::TheSentence=(); #Satz-Array bereit für neuen Satz
    }
}

#alle Dateien schliessen
close NEGRA;
close POS;
close NEG;
close LOG;
#Programm-Ende

#####
# Unterroutine Annotate:
# fügt allen temporalen Ausdrücken ein zusätzliches Merkmal %%TEMP_n hinzu
# (wobei n den Temporal-Wert angibt)
#####

sub Annotate {
    my $Line;
    my $LineNr=0;                #Nummer des Tokens (Zeilen-Nr.)

    #Counter, der benötigt wird, wenn ein neuer Parent-Knoten eingeführt
    #werden muss (cp. Unterroutine PrevCheck)
    my $NewParentCounter=0;

    #zeilenweises (tokenweises) Verarbeiten des Satzes
    foreach $Line(@Tempex::TheSentence) {
        if($Line=~m/#.OS.*){ $LineNr++; next;}
        my @Token=split(/t/, $Line); #aufsplitten in die Token-Komponenten
        my $Node=$Token[0];          #Knoten-Tag
        my $Pos=$Token[1];           #Part of Speech-Tag
        my $Edge=$Token[3];          #Kanten-Tag
        my $Parent=$Token[4];        #Parent-Tag

        #Liste, die die von LexCheck zurückgelieferten Werte
        #TempValue und LokValue entgegennimmt
        my @ValueList=();

        #Variable bzw. Wert, die/der angibt, ob es sich bei dem Token um
        #einen (potentiell) temporalen Ausdruck handelt
        my $TempValue=0;

        #Variable bzw. Wert, die/der angibt, ob ein potentiell temporales
        #Token auch potentiell lokal ist
        my $LokValue=0;

        #-----#
        # Bestimmung des Wertes von $TempValue je nach Wortart
        # - ADV: Adverb
        # - ADJD: adverbiale od. prädikative Adjektiv
        # - PWAV: adverbiales Interrogativ- oder Relativpronomen
        # - PAV: Pronominaladverb
        # - APPR: Präposition, Zirkumposition links
        # - APPO: Postposition
        # - APZR: Zirkumposition rechts
        # - APPRART: Präpositon mit Artikel
        # - NN: Substantiv
        # - AVP: Adverbialphrase
        # - AP: Adjektivphrase
        # - NP: Nominalphrase
    }
}

```

```

# - PP: Präpositionalphrase
# - CAC: coordinated Apposition
# - CARD: Kardinalzahl
#-----#
SWITCH: {
  if($Pos eq 'ADV')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'ADJD')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'PWAV')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'PAV')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'APPR')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'APPO')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'APZR')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'APPRART')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'NN')
    {@ValueList=&LexCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'AVP')
    {$TempValue=&PhraseCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'AP')
    {@ValueList=&PhraseCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'NP')
    {@ValueList=&PhraseCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'PP')
    {@ValueList=&PhraseCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'CAC')
    {@ValueList=&PhraseCheck($Node, $Pos); last SWITCH;}
  if($Pos eq 'CARD')
    {@ValueList=&CardCheck($Node, $Pos); last SWITCH;}
} #Ende des SWITCH-Blocks

$TempValue=$ValueList[0];
$LokValue=$ValueList[1];

#-----#
# Erkennen von falsch geparsten Adverbial- und Präpositionalphrasen
#-----#
if(($Pos eq 'ADV')&&($TempValue ne '0')){
  my $NewParent=&PrevCheck($LineNr, $NewParentCounter);
  if($NewParent ne '0'){
    $Token[4]=$NewParent;
    $Token[3]='HD';
    $NewParentCounter++;
  }
}

#-----#
# Wenn nach einer Präposition nochmals eine Präposition folgt ->
# erste Präposition in PP aufnehmen
#-----#
if(($Pos eq 'APPR')&&($Node eq 'bis')){
  my $NewParent=&NextCheck($LineNr, $Pos);
  if($NewParent ne '0'){
    $Token[4]=$NewParent;
    $Token[3]='AC';
  }
}

#-----#
# Wenn vor einer temporalen PP ein Adverb steht ->
# falls das Adverb in prev_adv.lex vorkommt: Aufnahme in die PP,
# PP mind. Temporalwert 2
#-----#
if($Pos eq 'ADV'){
  my $PrevValue=&LexCheck($Node, 'PrevADV');
  my $NewParent=0;
  if($PrevValue ne '0'){
    $NewParent=&NextCheck($LineNr, $Pos);
    if($NewParent ne '0'){
      $Token[4]=$NewParent;
      $Token[3]='MO';
    }
  }
}

#-----#
# wenn das Token (potentiell) temporal ist, dann füge eine
# zusätzliche Komponente(Tag) unter Angabe des Wertes von
# $TempValue hinzu
#-----#
if($TempValue!=0){push(@Token, '%TEMP_', "$TempValue")}
if($LokValue!=0){push(@Token, '%LOK_', "$LokValue")}

#fügt die Komponenten des Tokens wieder zu einer Zeile zusammen

```

```

        $Line=join("\t", @Token);
        $LineNr++;
    }
    #Erhöht den Token-Counter um 1
}
#Ende der Unterroutine

#####
# Unterroutine PhraseCheck
# - Parameter 0: Knoten
# - Parameter 1: Wortart (Pos)
# sucht aus einer Phrase das für die Temporal-Bestimmung relevante Wort und übergibt es an LexCheck,
# liefert den Temporal-Wert der Phrase zurück (= Summe der Temporal-Wert der relevanten Komponenten
# der Phrase) - in einer Liste zus. mit Lokal-Wert
#####

sub PhraseCheck {

    my $Node=$_[0];          #die Nummer des Phrasen-Knotens
    $Node=~s/#!/;           #entfernt die Raute am Beginn der Knoten-Nummer
    my $Pos=$_[1];         #Wortart (Part of Speech) der Phrase
    my @ValueList=();       #die von Lex-Check zurückgelieferte Werte-Liste
    my $TempValue=0;        #Temporal-Wert der Phrase=Summe der Werte ihrer Komponenten
    my $LokValue=0;        #Lokal-Wert einer (potentiell) temporalen Phrase
    my $Line;               #Zeile(=Token) innerhalb des Satzes

    foreach (@Tempex::TheSentence) {
        $Line=$_;          #geht jede Zeile (jedes Token) des Satzes durch
        my @Token=split(/ /, $Line);
        #splittet die Zeile (das Token) in seine Bestandteile auf

        #-----#
        # Die relevanten Elemente der jeweiligen Phrase an LexCheck übergeben
        #-----#
        if(
            (($Pos eq 'AVP')&&($Token[4] eq $Node)&&($Token[3] eq 'AVC' || 'HD')) or
            (($Pos eq 'AP')&&($Token[4] eq $Node)&&($Token[3] eq 'HD')) or
            (($Pos eq 'NP')&&($Token[4] eq $Node)&&($Token[3] eq 'NK')) or
            (($Pos eq 'PP')&&($Token[4] eq $Node)&&($Token[3] eq 'AC')) or
            (($Pos eq 'PP')&&($Token[4] eq $Node)&&($Token[3] eq 'NK' || 'HD'))
        ){
            @ValueList=&LexCheck($Token[0], $Token[1]);
            $TempValue+=$ValueList[0];
            $LokValue+=$ValueList[1];
        }
    }
    #Ende der foreach-Schleife

    return ($TempValue, $LokValue);
}
#Ende der Unterroutine PhraseCheck

#####
# Unterroutine LexCheck
# - Parameter 0: Wort
# - Parameter 1: Wortart (Pos)
# überprüft ein Wort auf sein Vorkommen im entsprechenden Lexikon und gibt die Werte für
# TempValue und LokValue zurück
#####

sub LexCheck {
    my $Word=lc($_[0]);     #alles in Kleinbuchstaben
    my $Pos=$_[1];         #Part of Speech-Tag
    my $TempValue=0;
    my $LokValue=0;
    my %LexHash;

    #-----#
    # Festlegen des zugehörigen Lexikon-Hashs, in dem (je nach Wortart)
    # nachgeschaut werden soll
    #-----#
    POS: {
        #Adverbien
        if ($Pos eq 'ADV') {%LexHash=%AdvHash; last POS;}
        if ($Pos eq 'ADJD') {%LexHash=%AdvHash; last POS;}
        if ($Pos eq 'PWAV') {%LexHash=%AdvHash; last POS;}
        if ($Pos eq 'PAV') {%LexHash=%AdvHash; last POS;}
        if ($Pos eq 'PrevADV') {%LexHash=%PrevAdvHash; last POS;}

        #Präpositionen
        if($Pos eq 'APPR') {%LexHash=%PraepHash; last POS;}
        if($Pos eq 'APPO') {%LexHash=%PraepHash; last POS;}
        if($Pos eq 'APZR') {%LexHash=%PraepHash; last POS;}
        if($Pos eq 'APPRART') {%LexHash=%PraepHash; last POS;}

        #Substantive
        if ($Pos eq 'NN') {%LexHash=%NnHash; last POS;}
    }
    #Ende des POS-Blocks

    #-----#
    # Lookup im Lexikon-Hash und Speicherung der Temp- und Lok-Werte
    #-----#
}

```

```

LOOKUP: {
    my $Entry;

    # präzises Matching: das Wort kommt als Schlüssel im Hash vor:
    if (exists $LexHash{$Word}) {
        $TempValue=$LexHash{$Word}[0];
        $LokValue=$LexHash{$Word}[1];
    }
    #Ende des if-Blocks

    # ungenaues Matching: der hintere Teil des Wortes kommt
    # in der Liste alles Schlüssel des Lexikon-Hashs vor:
    my @Entries=keys %LexHash;
    foreach $Entry (@Entries) {
        if($Word=~m/.*$Entry\b/) {
            $TempValue=$LexHash{$Entry}[0];
            $LokValue=$LexHash{$Entry}[1];
        }
        #Ende des if-Blocks
    }
    #Ende der while-Schleife

}
#Ende des LOOKUP-Blocks

return ($TempValue, $LokValue);
#Ende der Unterroutine LexCheck
}

#####
# Unterroutine LexToHash
# - Parameter 0: Lexikon-Datei
# - Parameter 1: Referenz auf den Lexikon-Hash
# wandelt die Lexikon-Datei in einen Lexikon-Hash um
#####

sub LexToHash {
    my $Lex=$_[0];
    my $HashRef=$_[1];

    open(LEX, "$Lex") or die "Das Lexikon $Lex konnte nicht geöffnet werden in LexToHash";
    while(<LEX>) {
        #bearbeitet jede Zeile (jeden Eintrag)
        my $TheLine=$_;
        chomp($TheLine);
        my @Entry=split(/\t/, $TheLine);
        #liest den Eintrag in eine Liste ein
        my $Word=$Entry[0];
        #erstes Element des Eintrags=das Wort
        my $Values={$Entry[1], $Entry[2]};
        #Referenz auf die Werteliste

        #-----#
        # fügt dem Hash, auf den $HashRef verweist, einen neuen Datensatz mit
        # Schlüssel=$Word und Wert=Referenz auf die Werteliste hinzu.
        #-----#
        $$HashRef{"$Word"}=$Values;
    }
    #Ende der while-Schleife
    close LEX;
}
#Ende der Unterroutine LexToHash

#####
# Unterroutine PrevCheck
# - Parameter 0: Nummer des aktuellen Satzes
# - Parameter 2: Nummer des zuletzt eröffneten Parent-Knotens
# überprüft, ob das einem temporalen Adverb vorangehende Token mit dem Adverb zu einer Phrase (AVP
# resp. PP) zusammengefasst werden soll, und gibt die Nummer des neuen Parent-Knotens zurück
#####

sub PrevCheck {
    #Variablen-Initialisierung
    my $TheLineNr=$_[0];
    my $PrevLineNr= $TheLineNr-1;
    my $PrevLine=$Tempex::TheSentence[$PrevLineNr];
    my @PrevToken=split(/\t/, $PrevLine);
    my $PrevNode=$PrevToken[0];
    my $PrevPos=$PrevToken[1];
    my $PrevFeature=0;
    my $NewParent=0;
    my $NewParentCounter=$_[1];

    #-----#
    # Bedingung, unter denen ein vorangehendes Token zum temporalen Adverb gezählt wird
    #-----#

    #wenn es ein Adverb aus dem PrevAdv-Lexikon ist
    if($PrevPos eq 'ADV'){ $PrevFeature=&LexCheck($PrevNode, 'PrevADV')}

    #wenn es eine Präposition ist
    if($PrevPos eq 'APPR'){ $PrevFeature=1}

    #wenn es die Negation "nicht" ist (nur wenn die entsprechende Programm-Option aktiviert ist)
    if(($NegOption eq 'y')&&($PrevPos eq 'PTKNEG')){ $PrevFeature=1}

    #-----#

```

```

# Erstellen einer Adverbial-Phrase AVP
#-----#
if ($PrevFeature ne '0') {
    $NewParent=600+$NewParentCounter;          #eröffnet eine neue Knoten-Nummer

    #-----#
    # Das vorangehende Token wird mit einem neuen Parent- und einem neuen Edge-Tag versehen
    #-----#
    $PrevToken[4]=$NewParent;                  #weist dem vorangehenden Wort das neue Knoten-Tag zu
    EDGES: {                                    #weist dem vorangehenden Wort das neue Kanten-Tag zu
        if($PrevPos eq 'ADV'){ $PrevToken[3]='MO'}
        if($PrevPos eq 'APPR'){ $PrevToken[3]='AC'}
        if($PrevPos eq 'PTKNEG'){ $PrevToken[3]='MO'}
    } #Ende des EDGES-Blockes
    $Tempex::TheSentence[$PrevLineNr]=join("\t", @PrevToken);    #fügt das korrigierte Token wieder in den Satz ein

    #-----#
    # Ein neuer Phrasen-Knoten (AVP resp. PP) wird in den Satz eingefügt
    #-----#
    NEWPARENT: {
        my $NewLine;
        if($PrevPos eq 'ADV'){ $NewLine="$NewParent\tAVP\t--\t--\t0"}
        if($PrevPos eq 'PTKNEG'){ $NewLine="$NewParent\tAVP\t--\t--\t0"}
        if($PrevPos eq 'APPR'){ $NewLine="$NewParent\tPP\t--\t--\t0"}
        my $EosLine=pop(@Tempex::TheSentence); #entfernt die EOS-Zeile vorübergehend vom Satz
        push(@Tempex::TheSentence, $NewLine);  #fügt die neue Zeile hinten an den Satz an
        push(@Tempex::TheSentence, $EosLine);  #fügt die EOS-Zeile wieder hinten an den Satz an
    }
    #Ende des if-Blocks
}

return $NewParent;
#Ende der Unterroutine PrevCheck

#####
# Unterroutine NextCheck
# - Parameter 0: Nummer des aktuellen Tokens
# - Parameter 1: Part of Speech-Tag
# überprüft das auf ein Token folgende Token ($NextToken) und gibt - falls Überprüfung positiv -
# dessen Knoten-Nummer zurück
#####

sub NextCheck {
    my $TheLineNr=$_[0];
    my $Pos=$_[1];
    my $NextLineNr=$TheLineNr+1;
    my $NextLine=$Tempex::TheSentence[$NextLineNr];
    my @NextToken=split(/\t/, $NextLine); print LOG $NextLine;
    my $NextNode=$NextToken[0];
    my $NextPos=$NextToken[1];
    my $NextParent=$NextToken[4];
    my $NewParent=0;

    #-----#
    # Bedingungen, unter denen das vorangehende Token zum darauffolgenden Phrase gehört
    #-----#
    # wenn eine Präposition (bis) vor einer anderen Präposition d.h. PP steht
    if(($Pos eq 'APPR')&&($NextPos eq 'APPR' || 'APPRART')){ $NewParent=$NextParent}
    # wenn ein - vorgängig als PrevAdv analysiertes - Adverb vor einer Präposition d.h. PP steht
    if(($Pos eq 'ADV')&&($NextPos eq 'APPR' || 'APPRART')){ $NewParent=$NextParent}

    return $NewParent;
}
#Ende der Unterroutine PrevCheck

#####
# Unterroutine CardCheck
# - Parameter 0: Wort bzw. Zahl
# - Parameter 1: Wortart (Pos)
# überprüft, ob es sich bei der Zahl um eine Jahresangabe handeln könnte
#####

sub CardCheck {
    my $Card=$_[0];
    my $Pos=$_[1];
    my $TempValue=0;
    my $LokValue=0;

    #bei 4-stelligen Zahlen mit 1 am Anfang: $TempValue um 1 erhöhen
    if($Card=~m/^(1)d{3}d{1}$){ $TempValue++;}
    #bei 4-stelligen Zahlen mit 19 oder 20 am Anfang: $TempValue nochmals um 1 erhöhen
    if($Card=~m/^(20|19)d{3}d{1}$){ $TempValue++;}

    return ($TempValue, $LokValue);
}
#Ende der Unterroutine CardCheck

```