形態素解析ツール ： 英語とTreeTaggerを中心に

田中, 省作
九州大学情報基盤センター研究部外国語情報メディア研究部門

https://doi.org/10.15017/1470502

出版情報：九州大学情報基盤センター広報 ： 学内共同利用版. 2 (2), pp.108-118, 2002-07. 九州大学情報基盤センター
バージョン：
権利関係：

形態素解析ツール
— 英語と TreeTagger を中心に —

田中省作*

1　はじめに

コンピュータで自然言語を処理する研究 (自然言語処理: 以後, 言語処理と書きます) も歴史を重ね, 現在, 仮名漢字変換などを代表に, その成果が様々な場面で使用されています. 実際, 後述する形態素解析, 統語解析のレベルであれば, 実用的なツールが公開され, 場合によってはフリーで使用することができるようになってきました. そこで, 本稿では, 言語処理の最も基礎的な要素技術である形態素解析についてお話し, その具体的なツールとして TreeTagger を紹介します.

2　形態素解析の概略

本節では, 形態素解析の言語処理における位置づけを簡単に説明します. 既に形態素解析を知っている方, ツールに興味がある方は, 本節を読み飛ばしても構いません.

言語処理における一般的な解析では, 英文や日本語文といった言語表現を段階的に解析を進めます. 大雑把に書くと,

1. 形態素解析
2. 統語解析 (構文解析)
3. 意味解析

となります. 解析を複合的に進めることもあります.

このうち形態素解析 (morphological analysis) は, 入力文中の単語とその品詞を同定し, 原形を求める作業です. 例えば, "She likes a cake or something like that." に対して,

> She/代名詞 likes/他動詞 (like) a/冠詞 cake/普通名詞
> or/接続詞 something/代名詞 like/前置詞 that/代名詞 ./記号

といった情報が得られることになります.

その後, 統語解析 (syntactic analysis) によって, 句間の修飾/被修飾関係といった統語的関係を明らかにします. 先の例であれば, 「"She" は "likes" の主語」, 「"a cake" と "something like that" が並列で名詞句となり, "likes" の目的語」といった情報が得られることになります.

これらの情報を基に文の意味を形式的な表現で与えるのが意味解析 (semantic analysis) です.

このように形態素解析は, 言語解析で最初に適用される最も基礎的な要素技術といえます. 現在, 形態素解析は, 既に実用的な精度に達しています[1]. 実際, 形態素解析は単体でも様々な場面

*九州大学 情報基盤センター 研究部 外国語情報メディア研究部門
E-mail : sho@cc.kyushu-u.ac.jp

[1]しかしながら, まだ幾つかの問題が残されているのも確かです. 例えば, 固有名詞や派生語や複合語などをより正確に認定できれば, さらなる精度向上が期待されます.

で活用されています. WWW における検索エンジンなどでも, より正確な Web ページの特徴付けに形態素解析が利用されています.

形態素解析について日本語と英語の場合を例に, もう少し詳しく見てみましょう. その作業は, 主に,

1. 単語分割 (word segmentation)
2. 品詞タグ付け (part of speech tagging)

に分けられます.

単語分割とは, 文中の単語を同定する作業です. 英語では文中の各単語を「分かち書き」しますので, あまり大きな問題にはなりません. 一方, 日本語では分かち書きしませんので, まず文中のどこからどこまでが一単語なのかという切れ目を入れなければなりません. 例えば, 「今日は日曜日だけど学校に行く」は,

今日 | は | 日曜日 | だ | けど | 学校 | に | 行く | .

と単語分割されます.

品詞タグ付けとは各単語の品詞を同定する作業です. 日本語に比べ英語は, 多品詞語が多く, 品詞同定は難しいといえます. それで, 英語で形態素解析というと, この品詞タグ付けを指す場合もあります. さきの "She likes a cake or somthing like that." の例を考えてみましょう. "like" という単語は, 自動詞・他動詞・副詞・形容詞・前置詞と多様な品詞として使うことができます. 最初の "likes" は他動詞, 後の "like" は前置詞として使われており, 品詞タグ付けでは, これらの差異をきちんと同定しなければなりません. また, 日本語での品詞タグ付けは英語のそれに比べれば比較的容易です. さきの例の「今日は日曜日だけど学校に行く」では, 単語分割後の結果に対し,

今日/名詞 | は/格助詞 | 日曜日/名詞 | だ/助動詞 | けど/接続助詞
| 学校/名詞 | に/格助詞 | 行く/動詞 | ./句点

といった品詞タグ付けがされることになります.

英語・日本語共に品詞タグ付けの後, 各単語の原形は容易に得られることになります.

3 形態素解析システム TreeTagger

英語は, 事実上の国際語であり, 英語を対象とした言語処理ツールは多くのものが提案されています. 本稿では, ドイツ Stuttgart 大学の Helmut Schmidt 博士が開発, 公開している統計的手法を用いた形態素解析システム TreeTagger を解説します.

3.1 TreeTagger のインストール

TreeTagger は実行形式で配布されていますので, 特にコンパイラなどを準備する必要はありません. プラットフォームとしては Sun/Solaris と Linux 用, トライアル版として Windows 用のものが準備されています. 本稿では, Linux 用を例に説明していきます.

まず, TreeTagger 用のディレクトリを適当な場所に作成します. ここでは, そのディレクトリを$TREETAGGER として説明します. 適宜, 読み換えて下さい. そして[2],

```
http://www.ims.uni-stuttgart.de/projekte/corplex/⇒
                        TreeTagger/DecisionTreeTagger.html
```

から, それぞれ,

- tree-tagger-linux-3.1.tar.gz (TreeTagger 本体)
- tree-tagger-scripts.tar.gz (各種スクリプト)
- english-par-linux-3.1.bin.gz (パラメタ・ファイル[3])
- install-tagger.sh (インストール用のスクリプト)

を$TREETAGGER ディレクトリにダウンロードします[4].

次に, $TREETAGGER ディレクトリでinstall-tagger.sh を実行します.

```
% sh install-tagger.sh

Linux version of TreeTagger installed.
Tagging scripts installed.
English parameter file (Linux) installed.
Path variables modified in tagging scripts.

You should add $TREETAGGER/cmd and $TREETAGGER/bin to the command search path.
```

$TREETAGGER/cmd と$TREETAGGER/bin をコマンドパスに加えておけば, どこからでも TreeTagger を実行できるようになります.

3.2　解析してみましょう

TreeTagger のタグ付けプログラムは, bin ディレクトリのtree-tagger です. このプログラムを直接呼び出すには, 入力フォーマットなどに幾つかの制約があり繁雑です[5]. 実際には, cmd ディレクトリのスクリプトを呼ぶようにします. 英語テキストの形態素解析の場合は,

```
tree-tagger-english 《英語テキストのファイル名》
```

で呼び出します. 例えば, "She likes a cake or something like that." という内容のファイルtest.dat 中の文を形態素解析する場合は, 次のようになります.

[2]誌面の都合上, '⇒' で折り返しています.

[3]TreeTagger は, Web ページに "a language independent part-of-speech tagger" とあり, このパラメタ・ファイルを入れ替えることで多言語に適用することができます. 現在同ページで, 英語・ドイツ語・フランス語・イタリア語のパラメタ・ファイルが準備されています. また, このパラメタファイルを同梱のパラメタ作成プログラム (tree-train) で作成すれば, 他言語にも適用できます.

[4]2002 年 5 月の時点で, TreeTagger の最新バージョンは 3.1 です.

[5]例えば, 一単語が一行に入力される必要がある, といったものです.

```
% cat test.dat
She likes a cake or something like that.

% tree-tagger-english test.dat
        reading parameters ...
        tagging ...
She PP she
likes VBZ like
a DT a
cake NN cake
or CC or
something NN something
like IN like
that DT that
. SENT .
         done.
```

reading parameters ...,tagging ...,done. は, 標準エラー出力へのメッセージで, 標準出力に解析結果が出力されます. 解析結果は, 一行に一単語に関する情報が次のような形式で表示されます. ただし, $(x)_{16}$ は, 16 進数表現のコード x であることを表します[6].

表記 $(09)_{16}$	品詞 $(09)_{16}$	原形

品詞は, Penn TreeBank Project[7] のタグセットを用いています. 記号とその意味を表 1 に挙げておきます. 例の解析結果を見ると, 最初の "like" は動詞 (正確には, VBZ, 3 人称単数現在形の動詞), 後の "like" は前置詞 (IN) として, きちんと認識されていることが分かりますね.

CC	Coordinatging conjunction	CD	Cardinal number
DT	Determiner	EX	Existential *there*
FW	Foreign word	IN	Preposition or subordinating conjunction
JJ	Adjective	JJR	Adjective, comparative
JJS	Adjective, superlative	LS	List item marker
MD	Modal	NN	Noun, singular or mass
NNS	Noun, plural	NP	Proper noun, singular
NPS	Proper noun, plural	PDT	Predeterminer
POS	Possessive ending	PP	Personal pronoun
PP$	Possessive pronoun	RB	Adverb
RBR	Adverb, comparative	RBS	Adverb, superlative
RP	Particle	SYM	Symbol
TO	*to*	UH	Interjection
VB	Verb, base form	VBD	Verb, past tense
VBG	Verb, gerund or present particle	VBN	Verb, past particle
VBP	Verb, non-3rd person singular present	VBZ	Verb, 3rd person singular present
WDT	Wh-determiner	WP	Wh-pronoun
WP$	Possessive wh-pronoun	WRB	Wh-adverb

表 1: 品詞の記号と意味

[6] $(09)_{16}$ は HT(Horizontal Tab), いわゆる水平タブに対応しています. 解析結果の例では形態素・品詞・原形が, 単なる空白で区切られているように見えますが, 実際には水平タブで区切られています.

[7] http://www.cis.upenn.edu/~treebank/

4　その他の代表的な形態素解析ツール – 日本語も含めて

TreeTagger 以外の代表的な形態素解析ツールをピックアップしておきます [2]. 使用条件やプラットフォームについては, Web ページを参照して下さい[8]. 英語については,

- Eric Brill's Tagger
  `http://www.cs.jhu.edu/~brill/`
- QTAG
  `http://www-clg.bham.ac.uk/QTAG/`
- TnT
  `http://www.coli.uni-sb.de/~thorsten/tnt/`

などがあります.

日本語では,

- 茶筅
  `http://chasen.aist-nara.ac.jp/`
- JUMAN
  `http://pine.kuee.kyoto-u.ac.jp/nl-resource/juman.html`
- ALTJAWS
  `http://www.kecl.ntt.co.jp/icl/mtg/resources/altjaws.html`
- Breakfast
  `http://www.labs.fujitsu.com/free/breakfast/`
- すもも
  `http://www.t.onlab.ntt.co.jp/sumomo/`

などがあります.

また, 言語処理ツールを体系的にリストアップしようとしている組織があり, NLSR(The Natural Language Software Resitry) といいます. NLSR には, 形態素解析だけでなく, 様々なレベルの言語処理ツールが登録されています. Web ページから, それらを検索できるようになっています. 興味がある方は, ぜひ一度訪れてみるとよいでしょう. URL は,

`http://registry.dfki.de/`

です.

5　形態素解析の応用

形態素解析は, 言語解析[9]における入口に相当する技術です. 一般の言語解析では, その結果を受けてさらに深い処理, 統語解析, 意味解析と進むことになります. ですが, 実際には, 形態素解析単体でも様々な場面で用いられています. ここでは簡単にその応用例を 2 つ紹介します.

[8]本稿で挙げた URL は, 2002 年 5 月に確認したものです.
[9]言語処理は主に, 文の構造を解析する言語解析と, 文の生成を行う言語生成に大別されます.

5.1 辞書引き

我々が英文を読んでいて英和辞書で単語の意味を調べる場合, その表現そのままで直接引くとは限りません. 一般に辞書の見出し語は単語の原形で統一されており, 辞書引きの際は調べたい表現を原形に直す必要があります. 例えば, "The TreeTagger consits of two programs: ..." において, "consists" や "programs" は,

consists ⟶ consist
programs ⟶ program

と直して辞書引きすることになります[10]. 形態素解析を適用すれば, 品詞同定と原形が求まるので, 辞書引きを自動化することができます. 辞書引きの自動化ツールでは, 内部で形態素解析が利用されていることもあります.

一例として, 本稿で説明した TreeTagger を用いて英文を形態素解析し, 辞書引きを自動化し, 訳語を埋め込む Web ページを,

`http://kushida.cc.kyushu-u.ac.jp/~sho/english-tagger/`

に作成してみました. このページでは, まずフォーム部分に英語テキストを入力します[11](図 5.1). そして, 形態素解析へをクリックすると, TreeTagger による解析結果が表示されます (図 5.1). さらに, 辞書引きへをクリックすると, 辞書引きして訳語を埋め込んだ HTML ドキュメントが表示されます. 訳語が埋め込まれた英単語は, 青色で表示され, そこにマウスを当てるとサブウィンドウが開き訳語が表示されます (図 5.1).

ただし, この Web ページでは, 名詞・形容詞・副詞・動詞に対してのみ辞書引きを行います. Web ブラウザとしては, Internet Explorer, Mozilla, Opera で動作を確認しています[12].

英和辞書としては, EDR の開発した英日対訳辞書を使用しました [1].

5.2 自家製コーパスへのタグ付け

コーパス (corpus) とは, 用例集のことです. 最近は多くの電子化コーパスが作成され, 「コーパス」といえば, 電子化されたものを指すことも少なくありません[13]. さらに, 統語や意味に関する情報まで付与されたタグ付き電子化コーパスも作成されるようになり, 言語学や言語処理の研究において無くてはならない重要な資源になっています.

これらのコーパスは, 外国語学習や作文支援に有効であることも言われています[14]. しかし, 研究論文といった非常に限られた分野における作成支援には, 個別領域に依存したコーパスが必要となります [7]. というのは, 研究分野毎に特有の用語や言い回しが存在し, それらについては一般的なコーパスではカバーできないからです. 現時点では, 個別領域に依存したコーパ

[10]このように原形に戻すことを lemmatization といいますが, 初習外国語ではこの lemmatization さえ, ままならないという場合があります. そのため, 最近の学習辞典などでは, 不規則な活用をする単語については屈折形でも引けるような配慮がされているものも多いようです.

[11]Copy&Paste などでも OK です.

[12]Netscape Navigator では, 形態素解析, HTML ドキュメントの作成はできますが, タグの関係上, 訳語を埋めこんだ単語にマウスを当ててもサブウィンドウが開きません.

[13]本稿でも今後, 単に「コーパス」と書いた場合は 電子化したものを指すことにします.

[14]コーパスを利用する際に, コンコーダンサ (concordancer) と呼ばれる効率的に検索・提示するツールを頻繁に使用します. これらについても別の機会に紹介していきたいと思います

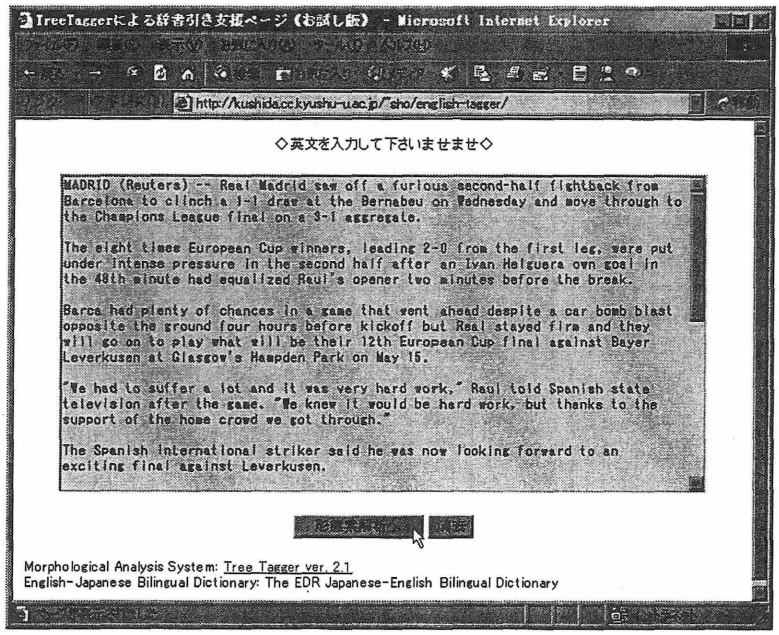
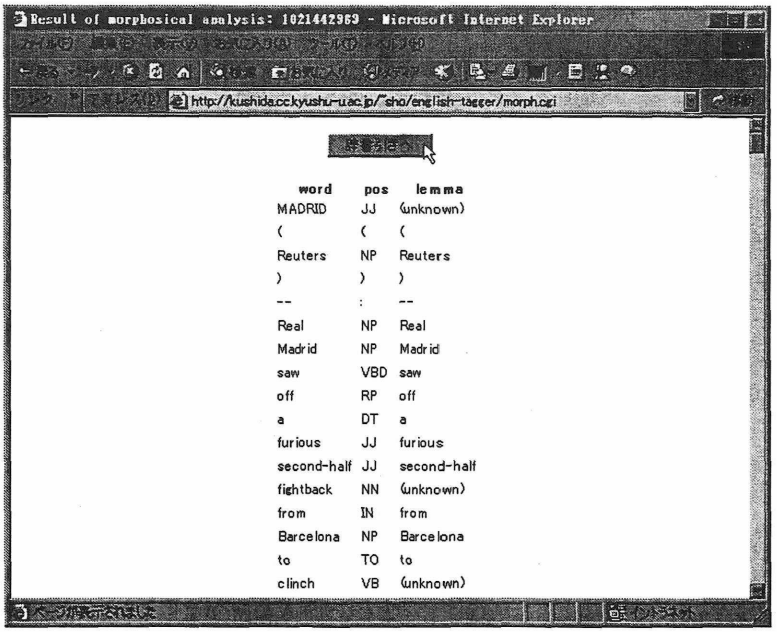


図 1: 英文テキストの入力



word	pos	lemma
MADRID	JJ	(unknown)
(	(	(
Reuters	NP	Reuters
)	)	)
--	:	--
Real	NP	Real
Madrid	NP	Madrid
saw	VBD	saw
off	RP	off
a	DT	a
furious	JJ	furious
second-half	JJ	second-half
fightback	NN	(unknown)
from	IN	from
Barcelona	NP	Barcelona
to	TO	to
clinch	VB	(unknown)

図 2: 形態素解析の結果

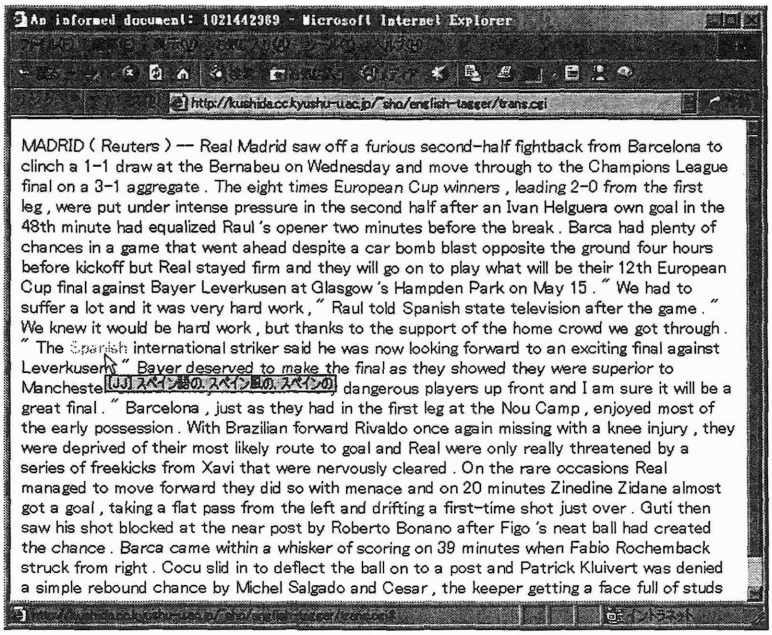


図 3: 訳語を埋め込んだ HTML ドキュメント (図では "Spanish" に埋めこまれた訳語が表示されています)

スについては, 各自で準備する他ありません. 実際, 多くの研究者が過去の大量の論文を蓄積し, 自分の論文作成の際に, それを参考書的に利用していると思います. 幸い近年, 論文や予稿集を PostScript や PDF という形に直し, CD-ROM やネットワーク上で公開・配布する国際会議や学会が増えています. これらをテキスト形式に変換して, テキスト・コーパスとして蓄えておけば, grep などの文字列検索コマンドを用いて容易に検索することが可能となります.

さらに, このテキスト・コーパスを形態素解析ツールでタグ付けすることによって, 語の屈折などを考慮せず, さらに品詞を組合せたより柔軟な検索が可能となります.

その他, Web ページの検索など, 情報検索でも形態素解析は重要な役割を果たしています. これに関しては文献 [3, 6] など詳しい入門書が出版されていますので, 本稿では割愛します.

6 おわりに

本稿では, 英語を中心に TreeTagger を例として形態素解析ツールを紹介しました. 普段我々が使っている「言葉」を処理するツールですから, 例えば, プログラミング言語や数値解析パッケージなどに比べれば, 身近に感じて頂けるのではないかな, と思います. ぜひ気軽にお試し下さい.

また, 機会があれば, これら形態素解析ツールだけでなく, より高位レベルの言語処理ツールなども紹介していきたいと思います.

謝辞

Helmut Schmidt 博士には, TreeTagger に関する筆者の初歩的な質問にも親切に解説して頂きました. 深謝致します.

参考文献

[1] 日本電子化辞書研究所: EDR 電子化辞書仕様説明書 (1995).

[2] 使いやすくなった自然言語処理のフリーソフト, 情報処理学会誌, Vol. 41, No. 11, pp. 1202-1238 (2000).

[3] 北 研二, 津田和彦, 獅々堀正幹: 情報検索アルゴリズム, 共立出版 (2002).

[4] 斉藤俊雄, 中村純作, 赤野一郎: 英語コーパス言語学, 研究社出版 (1998).

[5] 田中穂積監修: 自然言語処理 — 基礎と応用 —, 電子情報通信学会 (1999).

[6] 徳永健伸: 情報検索と言語処理, 東京大学出版 (1999).

[7] 外池俊幸: テキスト処理環境の整備: 作文・外国語学習支援, http://www.lang.nagoya-u.ac.jp/~tonoike/tono95.html (1995).

A　その他の外国語へのTreeTaggerの適用

TreeTaggerは, パラメタ・ファイルさえ変えれば, 英語以外の言語にも適用できます. 英語以外の他言語への適用について, ドイツ語を例に簡単に解説しておきます. ドイツ語では, 英語のアルファベットに加え, Ä,Ö,Ü,ä,ö,ü,ßを用います. TreeTaggerではISO8859-1というコード体系を仮定しています. ISO8859-1におけるドイツ語固有のアルファベットのコードを, 以下(表2)に示しておきます.

Ä	$(c4)_{16}$	ä	$(e4)_{16}$
Ö	$(d6)_{16}$	ö	$(f6)_{16}$
Ü	$(dc)_{16}$	ü	$(fc)_{16}$
ß	$(df)_{16}$		

表 2: ウムラウトおよびエスツェットのコード(ISO8859-1)

最初から, このコード体系で作成されたドイツ語テキストや, 何らかの入力手段があれば良いのですが[15], 代替してこれらのコードを直接入力せずÄ,Ö,Ü,ä,ö,ü,ßを\A,\O,\U,\a,\o,\u,\sなどで代用することも多いようです. そのようなテキストを解析するには, TreeTaggerに渡す前に前処理としてPerlなどで対応コードに変換することになります. 例えば, Perlで,

```
#!/usr/local/bin/perl

while (<>) {
    s/\\A/\xc4/g;
    s/\\O/\xd6/g;
    s/\\U/\xdc/g;
    s/\\a/\xe4/g;
    s/\\o/\xf6/g;
    s/\\u/\xfc/g;
    s/\\s/\xdf/g;
    print;
}
```

という変換プログラムを通して, TreeTaggerに入力すれば良いことになります.

TreeTaggerは直接, タグ付けプログラムを呼び出すのではなくて, cmdディレクトリのドイツ語形態素解析用のスクリプトを使って,

```
tree-tagger-german 《ドイツ語テキストのファイル名》
```

と呼び出します.

以下は, \を使ってウムラウトやエスツェットを代替表現したテキスト(test.dat)を, 上記のPerlプログラム(gconv.pl)で変換, TreeTaggerで形態素解析した例です.

[15]Windowsであれば, 代表的言語のキーボード・ドライバがあり, インストールして, 切り替えれば入力できます. MacOS 9.xでは, 西ヨーロッパ諸言語については, キーボード配列を切り替えることで入力できます.

```
% cat test.dat
Europ\aische Geschichte

Europa gibt es nicht. Es mu\s erfunden werden. Immer aufs neue. Und
jedesmal sieht es anders aus. Da ist etwas, mit unbestimmten Grenzen
und ungewissem Inhalt. Eine Schar verschiedener V\olker mit
unterschiedlichen Sprachen unt Gewohnheiten. Religionen, die kommen
und sich ver\andern. Literaturen beziehen und so \Uberliferungen
schaffen.

% ./gconv.pl test.dat | tree-tagger-german
        reading parameters ...
        tagging ...
         done.
Europäische ADJA europäisch
Geschichte NN Geschichte
Europa NE Europa
gibt VVFIN geben
es PPER PPER
nicht PTKNEG nicht
. $. .
Es PPER es
muß VMFIN müssen
erfunden VVPP erfinden
werden VAINF werden
. $. .
Immer ADV immer
aufs APPRART aufs
neue ADJA neu
. $. .
...
```