**TnT - Tagger Statistik Bagian-of-Speech**

Thorsten Brants Saarland University Computational Linguistics D-66041 Saarbru¨cken, Germany thorsten@coli.uni-sb.de

In Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, April 29 – May 3, 2000, Seattle, WA.

**Abstraksi**

Trigram Abstrak (TnT) adalah penandaan paruh waktu statistik yang efisien. Bertentangan dengan klaim yang ditemukan di tempat lain dalam literatur, kami berpendapat bahwa tagger yang didasarkan pada model Markov melakukan setidaknya serta pendekatan saat ini lainnya, termasuk kerangka Entropy Maksimum. Perbandingan baru-baru ini bahkan menunjukkan bahwa TnT berkinerja lebih baik secara signifikan untuk korpora yang diuji. Kami menjelaskan model dasar TnT, teknik yang digunakan untuk menghaluskan dan untuk menangani kata-kata yang tidak dikenal. Selanjutnya, kami menyajikan evaluasi pada dua korpora.

**1 Pendahuluan**

Sejumlah besar sistem pemrosesan bahasa saat ini menggunakan tagger bagian-of-speech untuk pra-pemrosesan. Tagger memberikan tag part-ofspeech (unik atau ambigu) untuk setiap token dalam input dan meneruskan outputnya ke level pemrosesan berikutnya, biasanya parser. Selain itu, ada minat besar dalam penandaan bagian-ofspeech untuk proyek penjelasan corpus, yang menciptakan sumber daya linguistik yang berharga dengan kombinasi pemrosesan otomatis dan koreksi manusia.

Untuk kedua aplikasi, tagger dengan akurasi setinggi mungkin diperlukan. Perdebatan tentang paradigma mana yang memecahkan masalah penandaan part-of-speech yang terbaik belum selesai. Perbandingan pendekatan terbaru yang dapat dilatih tentang korpora (van Halteren et al., 1998; Volk dan Schneider, 1998) telah menunjukkan bahwa dalam kebanyakan kasus, pendekatan statistik (Cutting et al., 1992; Schmid, 1995; Ratnaparkhi, 1996) menghasilkan hasil yang lebih baik daripada penandaan berbasis negara, berbasis aturan, atau berbasis memori (Brill, 1993; Daelemans et al., 1996). Mereka hanya dilampaui oleh kombinasi dari berbagai sistem, membentuk "tag pemilih".

Di antara pendekatan statistik, kerangka kerja Entropi Maksimum memiliki posisi yang sangat kuat. Namun demikian, perbandingan independen baru-baru ini dari 7 taggers (Zavrel dan Daelemans, 1999) telah menunjukkan bahwa pendekatan lain bahkan bekerja lebih baik: model Markov dikombinasikan dengan teknik perataan yang baik dan dengan penanganan kata-kata yang tidak dikenal. Tagger ini, TnT, tidak hanya menghasilkan akurasi tertinggi, tetapi juga tercepat dalam pelatihan dan pemberian tag.

Perbandingan tagger disusun sebagai "pengujian kotak hitam": tetapkan tugas yang sama untuk setiap tagger dan bandingkan hasilnya. Makalah ini menjelaskan model dan teknik yang digunakan oleh TnT bersama dengan implementasinya.

Pembaca akan terkejut betapa sederhananya model yang mendasarinya. Hasil perbandingan tagger tampaknya mendukung maxime "yang paling sederhana adalah yang terbaik". Namun, dalam makalah ini kami mengklarifikasi sejumlah detail yang dihilangkan dalam publikasi utama sebelumnya mengenai penandaan dengan model Markov. Sebagai dua contoh, (Rabiner, 1989) dan (Charniak et al., 1993) memberikan ikhtisar yang baik tentang teknik dan persamaan yang digunakan untuk model Markov dan penandaan sebagian teknologi, tetapi mereka tidak terlalu eksplisit dalam detail yang dibutuhkan untuk mereka. aplikasi. Kami berpendapat bahwa bukan hanya pilihan model umum yang menentukan hasil tagger tetapi juga berbagai keputusan "kecil" tentang alternatif.

Tujuan dari makalah ini adalah untuk memberikan penjelasan rinci tentang teknik yang digunakan dalam TnT. Selain itu, kami menyajikan hasil tagger pada corpus NEGRA (Brants et al., 1999) dan Penn Treebank (Marcus et al., 1993). Hasil Penn Treebank yang dilaporkan di sini untuk pendekatan model Markov setidaknya setara dengan yang dilaporkan untuk pendekatan Entropi Maksimum dalam (Ratnaparkhi, 1996). Untuk perbandingan dengan penanda lain, pembaca disebut (Zavrel dan Daelemans, 1999).

**2 Arsitektur**

**2.1 The Underlying Model**

TnT menggunakan model Markov orde kedua untuk penandaan sebagian teknologi. Status model mewakili tag, output mewakili kata-kata. Probabilitas transisi tergantung pada status, sehingga pasangan tag. Probabilitas keluaran hanya tergantung pada kategori terbaru. Secara eksplisit, kami menghitung

argmax t1...tT "T Y i=1 P(ti|ti−1,ti−2)P(wi|ti)#P(tT+1|tT) (1)

untuk urutan kata tertentu w1 ... wT dengan panjang T. t1 ... tT adalah elemen dari tagset, tambahan

tag t − 1, t0, dan tT + 1 adalah penanda awal-urutan dan akhir-urutan. Menggunakan tag tambahan ini, meskipun berasal dari pemrosesan tanda baca yang belum sempurna, sedikit meningkatkan hasil penandaan. Ini berbeda dari formula yang disajikan dalam publikasi lain, yang hanya berhenti dengan "longgar" pada kata terakhir. Jika batas kalimat tidak ditandai di input, TnT menambahkan tag ini jika bertemu dengan salah satu [.!?;] Sebagai token.

Probabilitas transisi dan keluaran diperkirakan dari corpus yang ditandai. Sebagai langkah pertama, kami menggunakan probabilitas kemungkinan maksimum ˆ P yang berasal dari frekuensi relatif:

Unigrams: ˆ P(t3) = f(t3) /N (2)

Bigrams: ˆ P(t3|t2) = f(t2,t3) /f(t2) (3)

Trigrams: ˆ P(t3|t1,t2) = f(t1,t2,t3) /f(t1,t2) (4)

Lexical: ˆ P(w3|t3) = f(w3,t3) /f(t3) (5)

untuk semua t1, t2, t3 dalam tagset dan w3 dalam leksikon. N adalah jumlah total token dalam corpus pelatihan. Kami mendefinisikan probabilitas kemungkinan maksimum menjadi nol jika nominator dan penyebutnya nol. Sebagai langkah kedua, frekuensi kontekstual dihaluskan dan frekuensi leksikal diselesaikan dengan menangani kata-kata yang tidak ada dalam leksikon (lihat di bawah).

**2.2 Smoothing**

Trigram probabilitas yang dihasilkan dari corpus biasanya tidak dapat langsung digunakan karena masalah sparsedata. Ini berarti bahwa tidak ada contoh yang cukup untuk setiap trigram untuk memperkirakan probabilitas dengan andal. Selain itu, menetapkan probabilitas ke nol karena trigram yang sesuai tidak pernah terjadi dalam korpus memiliki efek yang tidak diinginkan. Ini menyebabkan probabilitas urutan lengkap diatur ke nol jika penggunaannya diperlukan untuk urutan teks baru, sehingga membuatnya tidak mungkin untuk menentukan urutan urutan yang mengandung probabilitas nol.

Paradigma smoothing yang memberikan hasil terbaik di TnT adalah interpolasi linier unigrams, bigrams, dan trigram. Oleh karena itu, kami memperkirakan probabilitas trigram sebagai berikut:

P(t3|t1,t2) = λ1 ˆ P(t3) + λ2 ˆ P(t3|t2) + λ3 ˆ P(t3|t1,t2) (6)

P adalah estimasi kemungkinan maksimum dari probabilitas, dan λ1 + λ2 + λ3 = 1, jadi P sekali lagi mewakili distribusi probabilitas.

Kami menggunakan varian independen konteks dari interpolasi linier, yaitu, nilai λ tidak bergantung pada trigram tertentu. Bertentangan dengan intuisi, ini menghasilkan hasil yang lebih baik daripada varian yang bergantung pada konteks. Karena masalah data-jarang, seseorang tidak dapat memperkirakan set λ yang berbeda untuk setiap trigram. Oleh karena itu, merupakan praktik umum untuk mengelompokkan trigram berdasarkan frekuensi dan memperkirakan kumpulan λ yang terikat. Namun, kami tidak mengetahui adanya publikasi yang telah menyelidiki pengelompokan frekuensi untuk interpolasi linier dalam penandaan sebagian bicara. Semua pengelompokan yang telah kami uji menghasilkan hasil yang paling setara dengan interpolasi linier yang tergantung pada konteks. Beberapa pengelompokan bahkan menghasilkan hasil yang lebih buruk. Pengelompokan yang diuji termasuk a) satu set λs untuk setiap nilai frekuensi dan b) dua kelas (frekuensi rendah dan tinggi) pada dua ujung skala, serta beberapa pengelompokan di antara dan beberapa pengaturan untuk mempartisi kelas.

Nilai λ1, λ2, dan λ3 diperkirakan dengan interpolasi yang dihapus. Teknik ini secara berturut-turut menghilangkan setiap trigram dari corpus pelatihan dan memperkirakan nilai terbaik untuk λs dari semua ngram lainnya di corpus. Mengingat jumlah frekuensi untuk satu, dua, dan trigram, bobot dapat sangat efisien dengan waktu pemrosesan linier dalam jumlah trigram yang berbeda. Algoritma ini diberikan pada Gambar 1. Perhatikan bahwa mengurangi 1 berarti memperhitungkan data yang tidak terlihat. Tanpa pengurangan ini, model akan melebihi data pelatihan dan umumnya akan menghasilkan hasil yang lebih buruk.

**2.3 Penanganan Kata-Kata Yang Tidak Diketahui**

Saat ini, metode penanganan kata-kata yang tidak dikenal yang tampaknya bekerja paling baik untuk bahasa yang terpengaruh adalah analisis yang cukup seperti yang diusulkan dalam (Samuelsson, 1993). Probabilitas tag diatur sesuai dengan akhir kata. Su ﬃ x adalah prediktor yang kuat untuk kelas kata, misalnya, kata-kata di bagian Wall Street Journal dari Penn Treebank yang diakhiri dengan mampu adalah kata sifat (JJ) dalam 98% kasus (misalnya modis, variabel), sisanya 2% adalah kata benda (mis. kabel, variabel).

Distribusi probabilitas untuk sufiks tertentu dihasilkan dari semua kata dalam set pelatihan yang berbagi sux yang sama dari beberapa panjang maksimum yang telah ditentukan sebelumnya. Istilah su ﬃ x seperti yang digunakan di sini berarti “urutan akhir karakter kata” yang belum tentu su ﬃ x bermakna secara bahasa.

Kemungkinan diperhalus oleh abstraksi yang berurutan. Ini menghitung probabilitas tag t diberikan huruf m terakhir li dari kata huruf n: P (t | ln − m + 1, ... ln). Urutan konteks yang semakin umum menghilangkan karakter su ﬃ x, sehingga P (t | ln − m + 2, ..., ln), P (t | ln − m + 3, ..., ln ), ..., P (t) digunakan untuk menghaluskan. Formula rekursi adalah

P(t|ln−i+1,...ln)

= ˆ P(t|ln−i+1,...ln) + θiP(t|ln−i,...,ln) 1 + θi (7)

set λ1 = λ2 = λ3 = 0

foreach trigram t1,t2,t3 with f(t1,t2,t3) > 0

depending on the maximum of the following three values:

case

f(t1,t2,t3)−1 f(t1,t2)−1 : increment λ3 by f(t1,t2,t3)

case

f(t2,t3)−1 f(t2)−1 : increment λ2 by f(t1,t2,t3)

case

f(t3)−1 N−1 : increment λ1 by f(t1,t2,t3)

end

end

normalize λ1,λ2,λ3

set λ1 = λ2 = λ3 = 0

untuk setiap trigram t1, t2, t3 dengan f (t1, t2, t3)> 0

tergantung pada maksimum dari tiga nilai berikut:

kasus

f (t1, t2, t3) −1 f (t1, t2) −1: peningkatan λ3 oleh f (t1, t2, t3)

kasus

f (t2, t3) −1 f (t2) −1: increment λ2 oleh f (t1, t2, t3)

kasus

f (t3) −1 N − 1: kenaikan λ1 oleh f (t1, t2, t3)

akhir

end normalisasi λ1, λ2, λ3

Gambar 1: Algoritma untuk menghitung bobot untuk interpolasi linear konteks-independen λ1, λ2, λ3 ketika frekuensi n-gram diketahui. N adalah ukuran korpus. Jika penyebut dalam salah satu ekspresi adalah 0, kami mendefinisikan hasil dari ekspresi itu menjadi 0.

untuk i = m ... 0, menggunakan estimasi kemungkinan maksimum ˆ P dari frekuensi dalam leksikon, bobot θi dan inisialisasi

P(t) = ˆ P(t). (8)

Estimasi kemungkinan maksimum untuk su su x panjang i berasal dari frekuensi korpus oleh

P(t|ln−i+1,...ln) = f(t,ln−i+1,...ln) f(ln−i+1,...ln) (9)

Untuk model Markov, kita memerlukan probabilitas kondisional terbalik P (ln − i + 1, ... ln | t) yang diperoleh oleh inversi Bayesian. Argumentasi termotivasi teoretis menggunakan deviasi standar probabilitas kemungkinan maksimum untuk bobot θi (Samuelsson, 1993).

Ini menyisakan ruang untuk interpretasi.

1) Kita harus mengidentifikasi nilai yang baik untuk m, su ﬃ x terpanjang yang digunakan. Pendekatan yang diambil untuk TnT adalah sebagai berikut: m tergantung pada kata yang dimaksud. Kami menggunakan su x terpanjang yang dapat kami temukan di set pelatihan (mis., Yang frekuensinya lebih besar dari atau sama dengan 1), tetapi paling banyak 10 karakter. Ini adalah pilihan yang ditentukan secara empiris.

2) Kami menggunakan pendekatan konteks-independen untuk θi, seperti yang kami lakukan untuk bobot kontekstual λi. Itu ternyata menjadi pilihan yang baik untuk mengatur semua toi ke standar deviasi probabilitas kemungkinan maksimum tanpa syarat dari tag di dalam corpus pelatihan, yaitu, kami menetapkan

θi = 1 s – 1 s X j=1 ( ˆ P(tj) − ¯ P)2 (10)

untuk semua i = 0 ... m - 1, menggunakan tagset dari tag s dan rata-rata

P =1 s s X j=1 ˆ P(tj) (11)

Ini biasanya menghasilkan nilai dalam kisaran 0,03 ... 0,10.

3) Kami menggunakan taksiran yang berbeda untuk kata besar dan kecil, yaitu, kami mempertahankan dua percobaan yang berbeda tergantung pada penggunaan huruf besar dari kata tersebut. Informasi ini meningkatkan hasil penandaan.

4) Kebebasan lain menyangkut pilihan kata-kata dalam leksikon yang harus digunakan untuk penanganan suf. Haruskah kita menggunakan semua kata, atau beberapa di antaranya lebih cocok daripada yang lain? Menerima bahwa kata-kata yang tidak dikenal kemungkinan besar jarang, orang dapat berargumen bahwa menggunakan kata-kata yang jarang dalam leksikon adalah perkiraan yang lebih baik untuk kata-kata yang tidak diketahui daripada menggunakan kata-kata yang sering dalam jumlah kata yang tidak biasa. Oleh karena itu, kami membatasi prosedur penanganan su ﬃ x pada kata-kata dengan frekuensi lebih kecil dari atau sama dengan beberapa nilai ambang batas. Secara empiris, 10 ternyata menjadi pilihan yang baik untuk ambang ini.

**2.4 Kapitalisasi**

Informasi tambahan yang ternyata berguna untuk proses disambiguasi untuk beberapa korpora dan taget adalah informasi kapitalisasi. Tag biasanya tidak informatif tentang penggunaan huruf besar, tetapi distribusi probabilitas tag di sekitar huruf besar berbeda dengan yang tidak menggunakan huruf besar. Efeknya lebih besar untuk bahasa Inggris, yang hanya mengkapitalisasi nama yang tepat, dan lebih kecil untuk bahasa Jerman, yang mengkapitalisasi semua kata benda.

Kami menggunakan bendera ci yang benar jika wi adalah kata kapital dan sebaliknya. Bendera ini ditambahkan ke distribusi probabilitas kontekstual. Dari pada

P(t3|t1,t2) (12)

Kita gunakan

P(t3,c3|t1,c1,t2,c2) (13)

dan persamaan (3) hingga (5) diperbarui sesuai. Ini sama dengan menggandakan ukuran tagset dan menggunakan tag yang berbeda tergantung pada kapitalisasi.

**2.5 Pencarian Balok**

Waktu pemrosesan algoritma Viterbi (Rabiner, 1989) dapat dikurangi dengan memperkenalkan pencarian balok. Setiap negara yang menerima δ nilai lebih kecil dari largest terbesar dibagi dengan beberapa nilai ambang batas θ dikecualikan dari pemrosesan lebih lanjut. Meskipun algoritma Viterbi dijamin untuk menemukan urutan keadaan dengan probabilitas tertinggi, ini tidak lagi benar ketika pencarian balok ditambahkan. Namun demikian, untuk tujuan praktis dan pilihan θ yang tepat, hampir tidak ada perbedaan antara algoritma dengan dan tanpa balok. Secara empiris, nilai θ = 1000 ternyata sekitar dua kali lipat kecepatan tagger tanpa mempengaruhi akurasi.

Tagger saat ini menandai antara 30.000 dan 60.000 token per detik (termasuk ﬁ le I / O) pada Pentium 500 yang menjalankan Linux. Kecepatan terutama tergantung pada persentase kata-kata yang tidak diketahui dan pada tingkat ambiguitas rata-rata.

**3 Evaluasi**

Kami mengevaluasi kinerja tagger dalam beberapa aspek. Pertama-tama, kami menentukan akurasi pemberian tag rata-rata lebih dari sepuluh iterasi. Akurasi keseluruhan, serta akurasi terpisah untuk kata-kata yang dikenal dan tidak dikenal diukur.

Kedua, kurva pembelajaran disajikan, yang menunjukkan kinerja ketika menggunakan pelatihan korpora dengan ukuran yang berbeda, dimulai dengan sesedikit 1.000 token dan mulai dengan ukuran seluruh korpus (dikurangi set tes).

Karakteristik penting dari penanda statistik adalah bahwa mereka tidak hanya menetapkan tag untuk kata-kata tetapi juga probabilitas untuk menentukan peringkat tugas yang berbeda. Kami membedakan penugasan yang dapat diandalkan dari yang tidak dapat diandalkan dengan hasil bagi penugasan terbaik dan kedua1. Semua tugas yang hasil bagi ini lebih besar dari beberapa ambang dianggap dapat diandalkan, yang lain tidak dapat diandalkan. Seperti yang akan kita lihat di bawah, akurasi untuk penugasan yang andal jauh lebih tinggi.

Tes dilakukan pada partisi korpora yang menggunakan 90% sebagai set pelatihan dan 10% sebagai set tes, sehingga data uji dijamin tidak terlihat selama pelatihan. Setiap hasil diperoleh dengan mengulangi percobaan 10 kali dengan partisi yang berbeda dan rata-rata hasil tunggal.

Dalam semua percobaan, set tes yang berdekatan digunakan. Alternatifnya adalah prosedur round-robin yang menempatkan setiap kalimat ke-10 ke dalam set tes. Kami berpendapat bahwa set tes yang berdekatan menghasilkan hasil yang lebih realistis karena artikel yang sama sekali tidak terlihat diberi tag. Dengan menggunakan prosedur round-robin, bagian-bagian dari sebuah artikel sudah terlihat, yang secara signifikan mengurangi persentase kata-kata yang tidak diketahui. Karena itu, kami berharap bahkan hasil yang lebih tinggi ketika menguji pada setiap kalimat ke-10 alih-alih serangkaian berdekatan 10%.

Di bawah ini, akurasi menunjukkan jumlah tag yang ditetapkan dengan benar dibagi dengan jumlah token dalam korpus yang diproses. Tagger diizinkan untuk menetapkan tepat satu tag untuk setiap token.

Kami membedakan akurasi keseluruhan, dengan mempertimbangkan semua token dalam korpus tes, dan akurasi terpisah untuk token yang diketahui dan tidak dikenal. Yang terakhir menarik, karena biasanya token yang tidak diketahui jauh lebih sulit untuk diproses daripada token yang dikenal, yang daftar tag yang valid dapat ditemukan dalam leksikon.

**3.1 Memberi tag pada corpus NEGRA**

Corpus NEGRA Jerman terdiri dari 20.000 kalimat (355.000 token) teks-teks surat kabar (Frankfurter Rundschau) yang diberi penjelasan dengan bagian-bagian teknik spek dan predikat-argumen (Skut et al., 1997). Ini dikembangkan di Universitas Saarland di Saarbru¨cken2. Sebagian dari itu ditandai di IMS Stuttgart. Evaluasi ini hanya menggunakan bagian dari anotasi wicara dan mengabaikan anotasi struktural.

Akurasi pemberian tag untuk corpus NEGRA ditunjukkan pada tabel 2.

Gambar 3 menunjukkan kurva pembelajaran dari tagger, mis., Akurasi tergantung pada jumlah data pelatihan. Durasi pelatihan adalah jumlah token yang digunakan untuk pelatihan. Setiap lama pelatihan diuji sepuluh kali, pelatihan dan set tes dipilih secara acak dan terpisah, hasilnya dirata-rata. Durasi pelatihan diberikan pada skala logaritmik.

Sungguh luar biasa bahwa akurasi penandaan untuk kata-kata yang dikenal sangat tinggi bahkan untuk korpora pelatihan yang sangat kecil. Ini berarti bahwa kami memiliki peluang bagus untuk mendapatkan tag yang tepat jika sebuah kata terlihat setidaknya satu kali selama pelatihan. Persentase rata-rata token yang tidak diketahui ditampilkan di baris bawah setiap diagram.

Kami mengeksploitasi fakta bahwa tagger tidak hanya menentukan tag, tetapi juga menetapkan probabilitas. Jika ada alternatif yang memiliki probabilitas "hampir" dari penugasan terbaik, alternatif ini dapat dipandang sebagai sama cocok. Gagasan "dekat" diungkapkan oleh jarak probabilitas, dan ini pada gilirannya dinyatakan oleh hasil bagi probabilitas. Jadi, jarak probabilitas tag tbest terbaik dan tag talt alternatif dinyatakan oleh p (tbest) / p (talt), yang beberapa nilai lebih besar atau sama dengan 1 karena penugasan tag terbaik memiliki probabilitas tertinggi.

Gambar 4 menunjukkan akurasi ketika memisahkan tugas dengan quotients lebih besar dan lebih kecil dari ambang batas (karenanya tugas yang dapat diandalkan dan tidak dapat diandalkan). Seperti yang diharapkan, kami menemukan akurasi untuk itu

Tabel 2: Akurasi penandaan part-of-speech untuk corpus NEGRA, rata-rata lebih dari 10 uji coba, pelatihan dan set tes terpisah. Tabel menunjukkan persentase token yang tidak diketahui, akurasi terpisah dan standar deviasi untuk token yang dikenal dan tidak dikenal, serta akurasi keseluruhan

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | percentage | known | | unknown | | overall | | |
|  |  | unknowns | acc. | σ | acc. | σ | acc. | σ | |
|  | NEGRA corpus | 11.9% | 97.7% | 0.23 | 89.0% | 0.72 | 96.7% | 0.29 |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**NEGRA Corpus: POS Learning Curve**

|  |
| --- |
| Accuracy |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| • | | • | |  |  |  | • | |  |  |  | ~~•~~ | |  |  |  |  | ~~•~~ | |  |  |  | • | |  |  |  | • | |  |  | • | |  | ~~•~~ | | |  |  |  | Overall |  |
| 90 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | |  | ◦ | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | | |  | ◦ | | | |  |  |  |  |  | min =78.1% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | max=96.7% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  | ◦ | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | • | Known |  |
|  |  |  |  |  |  |  | ◦ | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | min =95.7% |  |
| 70 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | ◦ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | max=97.7% |  |
| 60 | ◦ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | Unknown |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | min =61.2% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | max=89.0% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | 2 | |  |  | 5 | | |  | 10 | | | | |  | 20 | | | | | 50 | | | |  | 100 | | | |  | 200 320 500 | | | | | | | 1000 ×1000 Training Length | | | |  |
| 50.8 | | | 46.4 | | | | 41.4 | | | | 36.0 | | | | | | 30.7 | | | | | 23.0 | | | | | 18.3 | | | |  | 14.3 11.9 10.3 | | | | | | | 8.4 avg. percentage unknown | | | |  |

Gambar 3: Kurva pembelajaran untuk menandai NEGRA corpus. Set pelatihan ukuran variabel serta set uji 30.000 token dipilih secara acak. Pelatihan dan set tes terpisah, prosedur diulang 10 kali dan hasilnya dirata-rata. Persentase tidak diketahui untuk pelatihan 500k dan 1000k ditentukan dari ekstensi yang tidak ditandai.

**NEGRA Corpus: Accuracy of reliable assignments**

|  |
| --- |
| Accuracy |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | ⋄ | | ⋄ | | ⋄ ⋄ | | | ⋄ | |  |  | ⋄ ⋄ | | |  | ⋄ | |  | ⋄ | | ⋄ | Reliable |  |
| 99 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | ⋄ | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | min =96.7% |  |
|  |  |  |  |  |  |  | ⋄ | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | max=99.4% |  |
| 98 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | ⋄ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 97 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ⋄ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 96 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | |  | 2 | | 5 10 20 | | | | | | | 50 | | 100 | |  | 500 | | | 2000 | | | | | 10000 | | | | | threshold θ | |  |
|  | 100 | | | 97.9 | | 95.1 92.7 90.3 | | | | | | | 86.8 84.1 81.0 | | | | | 76.1 71.9 68.3 | | | | | | | | 64.1 62.0 | | | | | % cases reliable | |  |
|  |  | – | | 53.5 | | 62.9 69.6 74.5 | | | | | | | 79.8 82.7 85.2 | | | | | 88.0 89.6 90.8 | | | | | | | | 91.8 92.2 | | | | | acc. of complement | |  |

Gambar 4: Akurasi penandaan untuk NEGRA corpus ketika memisahkan tugas yang dapat diandalkan dan tidak dapat diandalkan. Kurva menunjukkan akurasi untuk penugasan yang andal. Angka-angka di garis bawah menunjukkan persentase penugasan yang dapat diandalkan dan akurasi set pelengkap (mis., Penugasan yang tidak dapat diandalkan).

Tabel 5: Akurasi penandaan part-of-speech untuk Penn Treebank. Tabel menunjukkan persentase token yang tidak diketahui, akurasi terpisah dan standar deviasi untuk token yang diketahui dan tidak diketahui, serta akurasi keseluruhan.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | percentage | known | | unknown | | overall | | |
|  |  | unknowns | acc. | σ | acc. | σ | acc. | σ | |
|  | Penn Treebank | 2.9% | 97.0% | 0.15 | 85.5% | 0.69 | 96.7% | 0.15 |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**Penn Treebank: POS Learning Curve**

|  |
| --- |
| Accuracy |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | |  |
| • | | • | | |  |  |  |  |  | • | | | • | |  |  |  |  | • | |  |  |  | • | |  |  |  | ~~•~~ | |  |  |  |  | • | | | |  |  |  |  |  | ~~•~~ | |  |  |  |  |  | • | | |  |  | Overall |  |
| 90 |  | | | | | | | | | | | |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | min =78.6% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | | |  |  |  |  |  |  |  | ◦ | | |  |  |  |  |  | ~~◦~~ | |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | | |  |  |  | ◦ | | |  |  | ◦ | | |  | ◦ | | | | |  |  |  |  |  |  |  |  |  | |  |  |  |  | • | max=96.7% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Known |  |
|  |  |  |  |  |  |  |  |  |  | ◦ | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | min =95.2% |  |
| 70 |  |  |  | ◦ | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ◦ | max=97.0% |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 | ◦ | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Unknown |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | min =62.2% |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| 50 | max=85.5% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | |  |
|  | | | |  | | | | | | |  | | |  | | | | | |  | | | | |  | | | | |  | | | | | |  | | | | | | | | |  | | | | | | | |  | | | |  |
|  | 1 |  | 2 | | |  |  | 5 | | | | |  | 10 | | | |  | 20 | | | | 50 | | | |  | 100 | | | | |  | 200 | | | | | | 500 | | | | | | | | |  |  | 1000 ×1000 Training Length | | | | | | |  |
| 50.3 | | | 42.8 | | | |  | 33.4 | | | | | | 26.8 | | | | | 20.2 | | | | 13.2 | | | | | 9.8 | | | | |  | 7.0 | | | | |  | 4.4 | | | | | | | |  |  | 2.9 | | | | | avg. percentage unknown | | |  |

Gambar 6: Kurva pembelajaran untuk menandai Penn Treebank. Set pelatihan ukuran variabel serta set uji 100.000 token dipilih secara acak. Pelatihan dan set tes terpisah, prosedur diulang 10 kali dan hasilnya dirata-rata.

**Penn Treebank: Accuracy of reliable assignments**

|  |
| --- |
| Accuracy |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | ⋄ | | ⋄ | | ~~⋄ ⋄~~ | | |  | ⋄ | |  | ⋄ ⋄ | | | ⋄ ~~⋄~~ | | | ⋄ | Overall |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 99 |  |  |  |  |  |  |  | ⋄ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | min =96.6% |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | ⋄ | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | max=99.4% |  |
| 98 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | ⋄ | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 97 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ⋄ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 96 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | |  | 2 | |  | 5 10 20 | | | | | | 50 | | 100 | |  |  | 500 | | 2000 | | | | 10000 | | | threshold θ | |  |
| 100 | | | 97.7 | | | 94.6 92.2 89.8 | | | | | | 86.3 83.5 80.4 | | | | | | 76.6 73.8 71.0 | | | | | | 67.2 64.5 | | | % cases reliable | |  |
|  | – | | 53.5 | | | 62.8 68.9 73.9 | | | | | | 79.3 82.6 85.2 | | | | | | 87.5 88.8 89.8 | | | | | | 91.0 91.6 | | | acc. of complement | |  |

Gambar 7: Akurasi penandaan untuk Penn Treebank saat memisahkan tugas yang dapat diandalkan dan tidak dapat diandalkan. Kurva menunjukkan akurasi untuk penugasan yang andal. Angka-angka di garis bawah menunjukkan persentase penugasan yang dapat diandalkan dan keakuratan set pelengkap.

penugasan yang dapat diandalkan jauh lebih tinggi daripada penugasan yang tidak bertanggung jawab. Perbedaan ini, misalnya, berguna untuk proyek anotasi selama proses pembersihan, atau selama pra-pemrosesan, sehingga tagger dapat mengeluarkan beberapa tag jika tag terbaik diklasifikasikan sebagai tidak dapat diandalkan.

**3.2 Menandai Penn Treebank**

Kami menggunakan Wall Street Journal sebagaimana tercantum dalam Penn Treebank untuk eksperimen kami. Anotasi terdiri dari empat bagian: 1) struktur bebas konteks ditambah dengan jejak untuk menandai gerakan dan konstituen dis-kontinu, 2) kategori phrasal yang dianotasi sebagai label node, 3) satu set kecil fungsi gram matic yang dianotasi sebagai ekstensi ke label node, dan 4) tag bagian-of-speech (Marcus et al., 1993). Evaluasi ini hanya menggunakan anotasi part-of-speech.

Bagian Wall Street Journal dari Penn Tree-bank terdiri dari kira-kira. 50.000 kalimat (1,2 juta singa token).

Akurasi pemberian tag untuk Penn Treebank ditunjukkan pada tabel 5. Gambar 6 menunjukkan kurva pembelajaran dari tagger, yaitu, keakuratan tergantung pada jumlah data pelatihan. Durasi pelatihan adalah jumlah token yang digunakan untuk pelatihan. Setiap lama pelatihan diuji sepuluh kali. Pelatihan dan set tes terpisah, hasilnya rata-rata. Durasi pelatihan diberikan pada skala logaritmik. Sedangkan untuk NEGRA cor-pus, akurasi penandaan sangat tinggi untuk token yang diketahui bahkan dengan sejumlah kecil data pelatihan.

Kami mengeksploitasi fakta bahwa tagger tidak hanya menghapus tag, tetapi juga menetapkan probabilitas. Gambar 7 menunjukkan akurasi ketika memisahkan tugas dengan quotients lebih besar dan lebih kecil dari ambang batas (karenanya tugas yang dapat diandalkan dan tidak dapat diandalkan). Sekali lagi, kami menemukan bahwa akurasi untuk tugas yang dapat diandalkan jauh lebih tinggi daripada untuk tugas yang tidak dapat diandalkan.

**3.3 Ringkasan Hasil Tagging Bagian-of-Speech**

Keakuratan penandaan part-of-speech rata-rata adalah antara 96% dan 97%, tergantung pada bahasa dan tagset, yang setidaknya setara dengan hasil mutakhir yang ditemukan dalam literatur, mungkin lebih baik. Untuk Penn Treebank, (Ratnaparkhi, 1996) melaporkan akurasi 96,6% menggunakan pendekatan Entropy Maksimum, pendekatan HMM kami yang jauh lebih sederhana dan lebih cepat menghasilkan 96,7%. Perbandingan ini perlu diperiksa ulang, karena kami menggunakan crossvalida sepuluh kali lipat dan rata-rata hasil sementara Ratnaparkhi hanya membuat satu tes berjalan.

Akurasi untuk token yang diketahui secara signifikan lebih tinggi daripada untuk token yang tidak diketahui. Untuk data surat kabar Jerman, hasilnya 8,7% lebih baik ketika kata itu terlihat sebelumnya dan karena itu ada dalam leksikon, daripada ketika itu tidak terlihat sebelumnya (97,7% vs 89,0%). Akurasi untuk token yang dikenal tinggi bahkan dengan sangat

sejumlah kecil data pelatihan. Hanya 1000 token yang cukup untuk mencapai akurasi 95% -96% untuk mereka. Penting bagi pemberi tag untuk melihat kata setidaknya satu kali selama pelatihan.

Stochastic taggers menetapkan probabilitas untuk tag. Kami mengeksploitasi probabilitas untuk menentukan keandalan penugasan. Untuk subset yang ditentukan selama pemrosesan oleh tagger, kami mencapai tingkat akurasi lebih dari 99%. Keakuratan set komplemen jauh lebih rendah. Informasi ini dapat, misalnya, dieksploitasi dalam proyek anotasi untuk memberikan perlakuan tambahan pada penugasan yang tidak dapat diandalkan, atau untuk menyampaikan ambiguitas terpilih ke langkah pemrosesan selanjutnya.

**4 Kesimpulan**

Kami telah menunjukkan bahwa tagger yang didasarkan pada model Markov menghasilkan hasil yang canggih, meskipun klaim yang bertentangan ditemukan dalam literatur. Sebagai contoh, penandaan model Markov yang digunakan dalam perbandingan (van Halteren et al., 1998) menghasilkan hasil yang lebih buruk daripada semua penandai lainnya. Menurut pendapat kami, alasan untuk klaim yang salah adalah bahwa algoritma dasar menyerahkan beberapa keputusan kepada implementor. Jumlah kebebasan yang agak besar tidak ditangani secara rinci dalam publikasi sebelumnya: penanganan awal dan akhir urutan, teknik pemulusan yang tepat, cara menentukan bobot untuk probabilitas konteks, detail tentang penanganan kata-kata yang tidak diketahui, dan cara menentukan bobot untuk kata-kata yang tidak diketahui. Perhatikan bahwa keputusan yang kami buat menghasilkan hasil yang baik bagi Jerman dan Corpus Inggris. Mereka melakukannya untuk beberapa korpora lain juga. Arsitektur tetap berlaku untuk berbagai macam bahasa.

Menurut perbandingan tagger saat ini (van Halteren et al., 1998; Zavrel dan Daelemans, 1999), dan menurut perbandingan hasil yang disajikan di sini dengan yang ada di (Ratnaparkhi, 1996), kerangka Maximum Entropy tampaknya menjadi satu-satunya yang lain pendekatan menghasilkan hasil yang sebanding dengan yang disajikan di sini. Ini adalah topik penelitian masa depan yang sangat menarik untuk menentukan keuntungan dari kedua pendekatan ini, untuk menemukan alasan akurasi tinggi mereka, dan untuk menemukan kombinasi yang baik dari keduanya. TnT tersedia secara bebas untuk universitas dan organisasi terkait untuk tujuan penelitian (lihat <http://www.coli.uni-sb.de/~thorsten/tnt>).

**Ucapan terima kasih**

Terima kasih banyak kepada Hans Uszkoreit atas dukungannya selama pengembangan TnT. Sebagian besar pekerjaan pada TnT dilakukan sementara penulis menerima hibah dari Deutsche Forschungsgemeinschaft di Graduiertenkolleg Kognitionswissenschaft Saarbru¨cken. Korporasi beranotasi besar adalah prasyarat untuk mengembangkan dan menguji sebagian penanda teknologi, dan memungkinkan pembuatan model bahasa berkualitas tinggi. Karena itu, saya akan melakukannya mengucapkan terima kasih kepada semua orang yang mengambil upaya untuk membubuhi keterangan Penn Treebank, Susanne Corpus, Stuttgarter Referenzkorpus, NEGRA Corpus, Verbmobil Corpora, dan beberapa lainnya. Dan, yang tak kalah pentingnya, saya ingin mengucapkan terima kasih kepada para pengguna TnT yang memberi saya laporan bug dan saran berharga untuk perbaikan.

**References**

Thorsten Brants, Wojciech Skut, and Hans Uszko-reit. 1999. Syntactic annotation of a German newspaper corpus. In Proceedings of the ATALA Treebank Workshop, pages 69–76, Paris, France.

Eric Brill. 1993. A Corpus-Based Approach to Lan-guage Learning. Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania.

Eugene Charniak, Curtis Hendrickson, Neil Ja-cobson, and Mike Perkowitz. 1993. Equations for part-of-speech tagging. In Proceedings of the Eleventh National Conference on Artificial In-telligence, pages 784–789, Menlo Park: AAAI Press/MIT Press.

Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In Proceedings of the 3rd Conference on Applied Natural Language Processing (ACL), pages 133–140.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. Mbt: A memory-based part of speech tagger-generator. In Proceedings of the Workshop on Very Large Corpora, Copenhagen, Denmark.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. Compu-tational Linguistics, 19(2):313–330.

Lawrence R. Rabiner. 1989. A tutorial on Hid-den Markov Models and selected applications in speech recognition. In Proceedings of the IEEE, volume 77(2), pages 257–285.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In Proceedings of the Conference on Empirical Methods in Nat-ural Language Processing EMNLP-96, Philadel-phia, PA.

Christer Samuelsson. 1993. Morphological tag-ging based entirely on Bayesian inference. In 9th Nordic Conference on Computational Lin-guistics NODALIDA-93, Stockholm University, Stockholm, Sweden.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In Helmut Feldweg and Erhard Hinrichts, editors, Lexikon und Text. Niemeyer, T¨ubingen.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97, Washington, DC.

Hans van Halteren, Jakub Zavrel, and Walter Daele-mans. 1998. Improving data driven wordclass tag-ging by system combination. In Proceedings of the International Conference on Computational Lin-guistics COLING-98, pages 491–497, Montreal, Canada.

Martin Volk and Gerold Schneider. 1998. Compar-ing a statistical and a rule-based tagger for ger-man. In Proceedings of KONVENS-98, pages 125– 137, Bonn.

Jakub Zavrel and Walter Daelemans. 1999. Eval-uatie van part-of-speech taggers voor het cor-pus gesproken nederlands. CGN technical report, Katholieke Universiteit Brabant, Tilburg.