Ampliación automática de corpus mediante la colaboración de varios etiquetadores *

Fernando Enríquez, José A. Troyano, Fermín Cruz y **F. Javier Ortega**
Dep. de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Avda. Reina Mercedes s/n
41012 Sevilla
fenros@lsi.us.es

Resumen: La disponibilidad de grandes corpus con texto etiquetado es un aspecto esencial en muchas tareas del procesamiento del lenguaje natural. El esfuerzo que se requiere para etiquetar manualmente este gran número de frases ha animado a los investigadores a crear aplicaciones automáticas para este trabajo. Nuestra propuesta representa un método para incrementar el tamaño de un corpus pequeño de manera totalmente automática o con un mínimo esfuerzo, hasta que adquiera el número deseado de frases. El contenido que se añade al corpus se obtiene de cualquier fuente como puede ser Internet, de la cual se puedan extraer frases sin etiquetar para ser analizadas. Si consideramos el pequeño corpus etiquetado como la semilla, nuestro método hace que evolucione hasta lograr el tamaño deseado. El proceso se basa en la opinión de varios etiquetadores mediante la técnica de *co-training* y de la aplicación de un segundo nivel de aprendizaje mediante *stacking*. Esta última será la técnica que nos servirá para decidir cuáles de las nuevas frases etiquetadas serán seleccionadas para pasar a formar parte del corpus.
Palabras clave: Generación de recursos, aprendizaje automático, combinación de sistemas

Abstract: The availability of extense tagged data corpus is an essential aspect in many NLP tasks. The effort required for tagging manually this large number of phrases has encouraged many researchers like us to create automatic applications for this issue. Our approach represents a completely automatic method (optionally applying a minimum effort) for enlarging an already existing corpus, so it acquires the desired number of tagged phrases. The extra content of the corpus will be obtained from any knowledge source like the web, from where we extract untagged sentences to be analyzed. Considering the initial small corpus as the seed, our method makes it evolve until it reaches the goal size. The process is based on several taggers using the co-training technique, achieving the results after a number of iterations and applying the stacking scheme for deciding which new tagged sentences must be incorporated to the new corpus.
Keywords: Resource generation, machine learning, system combination

1. Introducción

Hay múltiples técnicas en el campo del procesamiento del lenguaje natural que están basadas en corpus. Esto significa que dependen de la disponibilidad de grandes corpus etiquetados para poder funcionar y obtener buenos resultados. Este tipo de técnicas se aplica a todas las tareas del PLN como son la extracción de entidades con nombre, la desambiguación de significados o el etiquetado morfológico comúnmente conocido como etiquetado POS. Esta dependencia de estos recursos generalmente no supone ningún problema para los que trabajan en las tareas y en los idiomas más populares. Sin embargo, existen situaciones en las que la ausencia de

* Parcialmente financiado por el Ministerio de Educación y Ciencia (TIN2004-07246-C03-03).

grandes textos etiquetados se convierte en un serio obstáculo. En ocasiones porque la tarea es relativamente nueva y hay pocos recursos para ella, y otras veces porque el interés recae en un idioma que se sale del conjunto de idiomas utilizado por la mayoría, habiendo muy pocos textos etiquetados o ninguno. En estos casos es donde la generación automática de recursos nos puede ayudar, evitando tener que proceder manualmente al etiquetado de grandes volúmenes de textos, con el alto coste que esto conlleva.

A continuación mostramos la estructura que presenta el artículo de aquí en adelante. En la siguiente sección introduciremos las técnicas más comúnmente usadas para generar recursos de forma automática. En la sección tres detallaremos las características de nuestra solución y comentaremos los etiquetadores que hemos utilizado. La sección cuatro presenta los resultados de nuestros experimentos. Por último, en la sección cinco, expondremos las conclusiones y el trabajo futuro.

2. Técnicas de bootstrapping

Podemos encontrar diversas técnicas que han sido utilizadas para este objetivo en el pasado. Probablemente la más sencilla sea el método de *self-training* (Clark, 2003). Consiste en entrenar un etiquetador con el corpus actual con la intención de extender el corpus de entrenamiento con un conjunto de frases nuevas etiquetadas por el propio etiquetador. A continuación reentrenamos el etiquetador con el corpus extendido ejecutándose una nueva iteración. Este proceso se repite hasta que se obtenga el tamaño deseado o se alcance el número de iteraciones que se ha especificado a priori. Este es un esquema de muy bajo coste pero desafortunadamente la calidad del corpus decrece a un ritmo que hace inviable la obtención de un tamaño grande cumpliendo unos requisitos mínimos de calidad. La razón radica en la imposibilidad del etiquetador de aprender de él mismo, haciéndose cada vez más influyentes los errores que va cometiendo en las sucesivas iteraciones. Si existen situaciones que no aparecen en el corpus de entrenamiento, el etiquetador nunca aprenderá a responder de forma correcta ante estas situaciones sin recibir ayuda externa.

La técnica de *co-training* (Blum, 1998) permite que distintos etiquetadores puedan cooperar aportando así la ayuda externa de la que carece el *self-training*. Para aportar información adicional a un etiquetador sobre situaciones que no sabe manejar, se introduce otro etiquetador (o más). Cada etiquetador representa una visión diferente del problema, y la mezcla de estas vistas enriquece las capacidades globales del sistema (Abney, 2002). El proceso se describe a continuación:

- Cada etiquetador aprende de diferentes corpus aunque al comienzo, el corpus semilla será el mismo para todos y por lo tanto lo compondrán las mismas frases. De ahí en adelante cada corpus se ampliará de forma independiente, con diferentes frases y etiquetas.
- Cada etiquetador se ejecuta sobre un conjunto de frases nuevas sin etiquetar.
- Las frases que devuelve como resultado un etiquetador se añaden al corpus de entrenamiento del otro etiquetador.
- Reentrenamos los etiquetadores y repetimos el proceso.

El método colaborativo difiere de la técnica de *co-training* en el número de corpus de entrenamiento que se generan, existiendo en esta ocasión un único corpus para todos los etiquetadores. El aspecto principal será la selección de frases nuevas para extender el corpus inicial, teniendo que alcanzarse un acuerdo entre las opiniones de los diferentes etiquetadores involucrados en el sistema. Existen muchos métodos de selección que pueden implementarse, desde un sistema de votación a algo más complejo como el *stacking*, una técnica de aprendizaje automático que combina los resultados de una fase previa de aprendizaje (en nuestro caso los etiquetadores entrenados con el corpus de la iteración actual) con un segundo algoritmo de selección.

La última técnica que queremos mencionar antes de pasar a nuestra solución es la del *active learning*, que introduce una fase manual para etiquetar únicamente aquellas frases que presentan dudas para los sistemas automáticos que se están utilizando (en nuestro caso serán aquellas frases que producen desacuerdo entre los diferentes etiquetadores).

Nuestro sistema intentará tomar ventaja de la combinación de las diferentes vistas pre-

sentadas por la técnica de *co-training*, introduciendo un esquema por acuerdos mediante el esquema del *stacking*. También hemos estudiado los efectos que produce en los resultados la inserción de una fase manual tal y como propone el *active learning*.

3. Nuestra propuesta

Como base de nuestro modelo de colaboración, hemos seleccionado tres etiquetadores reentrenables de entre los más usados hoy en día. Estos son:

- *TnT* (Brants, 2000): Está basado en los modelos de Markov de segundo orden, calculando los trigramas para extraer las probabilidades de emisión de palabras y transición entre etiquetas. Incorpora algoritmos de suavizado y análisis de sufijos entre otros.
- *TreeTagger* (Schmid, 1994): Difiere de otros etiquetadores probabilísticos en la forma en que se calculan las probabilidades de transición, ya que utiliza árboles de decisión en un contexto de tetragramas.
- *MBT* (Daelemenas, 1996): Utiliza un método de aprendizaje basado en ejemplos. Estos algoritmos constituyen una forma de aprendizaje supervisado basado en el razonamiento por similitud, en donde la etiqueta para una palabra en su contexto se extrapola de los casos más similares procesados anteriormente.

También se probaron otros etiquetadores que finalmente no introducían mejoras en el sistema por diversas razones. En algunos casos no se comportaban bien al principio del proceso (con muy poca información de partida), otros introducían demasiado retardo en el sistema en cada iteración mermando en exceso la eficiencia temporal y otros no cumplían con el requisito de aportar una visión diferente del problema al estar basados en métodos similares a los ya presentes.

3.1. Co-training ingenuo

El primer punto a tratar es la creación de una línea base para comparar los resultados, implementando un sistema sencillo que aporte una solución con un bajo coste. Este sistema lo llamamos *naive* (ingenuo) *co-training* y se muestra en la figura 1. Tenemos tres etiquetadores para combinar y tres corpus que se utilizarán para entrenarlos. Cada corpus se extenderá introduciendo frases nuevas etiquetadas por los otros dos. Del conjunto de frases nuevas extraídas de la fuente se escogerá por tanto la mitad con el etiquetado propuesto por un etiquetador y la otra mitad con el propuesto por el segundo etiquetador, pasando a formar parte todas ellas del corpus de entrenamiento del tercero.

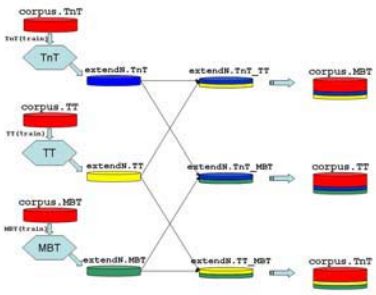

Figura 1: *Naive co-training* (tres etiquetadores).

En este esquema todas las frases nuevas se introducen en los corpus ya que no existe un criterio de selección entre ellas, provocando un crecimiento homogéneo y constante.

3.2. Stacking más co-training

Tras establecer un punto de partida con el *naive co-training*, hemos mejorado la precisión del corpus al final del proceso empleando un esquema de *stacking*, que aportará un método de arbitraje y consenso entre las diferentes opiniones introducidas por los dos etiquetadores que se combinan en cada ocasión. Este esquema ha sido aplicado con éxito en diversas tareas, en concreto en el ámbito del Procesamiento del Lenguaje Natural existen trabajos que aplican una doble etapa de decisión para el etiquetado POS (Halteren, 2001), la desambiguación de significados (Florian, 2002), el análisis sintáctico (Henderson, 1999) o el reconocimiento de entidades con nombre (Florian, 2003).

Dado que este sistema requiere más información para responder correctamente, aplicamos la fase de *stacking* tras cinco iteraciones del método *naive*, ya que resulta muy difícil para el *stacking* extraer información de las pocas frases que contiene el corpus semilla inicial.

Para generar la base de datos de entrenamiento del *stacking*, cogemos las frases etiquetadas desde el comienzo (las que inicialmente constituyeron el corpus semilla y que al comenzar la fase de stacking se extraen del corpus de entrenamiento). Estas frases, que contienen las etiquetas correctas ya que no han sido introducidas de forma automática por el sistema, constituyen el oráculo y se considera la fuente de conocimiento segura en la que se apoya el *stacking*.

Durante el proceso, cada etiquetador se ejecuta sobre el oráculo y sobre el conjunto de frases nuevas que se espera terminen extendiendo los corpus (extraídas en cada iteración de una fuente de frases sin etiquetar). El resultado aportado por dos etiquetadores sobre el oráculo, más las etiquetas reales de que disponemos, componen la base de datos de entrenamiento y el resultado de esos dos etiquetadores sobre el conjunto de frases nuevas constituyen la base de datos de aplicación. El proceso se ilustra en la figura 2.

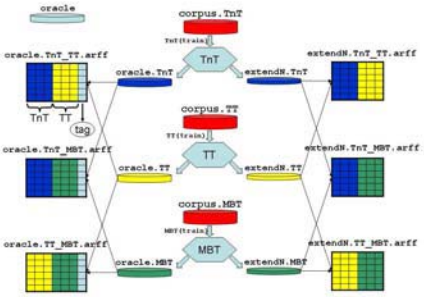

Figura 2: *Stacking* más *co-training* (tres etiquetadores).

Para aplicar la segunda fase de aprendizaje que representa la técnica de *stacking*, hacemos uso del sistema *weka* (Witten, 2000), que aporta una interfaz para aplicar algoritmos de aprendizaje automático ya incorporados en una librería muy extensa. Hemos configurado *weka* para utilizar un algoritmo de aprendizaje automático basado en árboles de decisión tras ejecutar diferentes algoritmos disponibles y comprobar que éste ofrecía los mejores resultados en la mayoría de las pruebas.

El árbol de decisión es construido a partir de la base de datos de entrenamiento, y es posteriormente usado para decidir las etiquetas de la base de datos de aplicación (que ha sido creada a partir de frases nuevas). El árbol proporciona para cada palabra la etiqueta más probable y su correspondiente probabilidad. Si esta probabilidad sobrepasa para cada palabra de una frase un umbral definido previamente, la frase es seleccionada para ampliar el corpus, y en caso contrario se rechaza.

3.3. La fase de revisión

El esquema que sigue el método *naive* es muy simple y muy rápido pero introduce gran cantidad de errores en los corpus que afectan en las sucesivas iteraciones a los resultados aportados por el sistema. Para intentar paliar este problema antes de que se aplique la fase de *stacking* que sigue a la de *naive*, hemos incorporado una fase intermedia que revisa las frases introducidas anteriormente corrigiendo en la medida de lo posible los errores cometidos. La fase consiste en repetir el etiquetado de todas las frases introducidas por el método *naive*, pero con el método *stacking*, aprovechando que existen más ejemplos que los disponibles en el corpus semilla del principio.

Aunque durante la revisión no se introducen frases nuevas (puede que incluso se descarten algunas que se habían introducido), los resultados a lo largo del resto del experimento se ven afectados de forma positiva, como comprobaremos en la siguiente sección.

Además de este tipo de revisión, también implementamos otro modo diferente que simula la participación de un experto etiquetando manualmente las palabras que generen más dudas al sistema durante esta fase. En los casos en los que la etiqueta no supere el umbral de fiabilidad especificado al comienzo del experimento, en lugar de ser rechazada se incorporará junto con la etiqueta real en lugar de la propuesta (es una simulación que podemos realizar al tener las frases nuevas etiquetadas en lugar de haber sido extraídas de una fuente sin etiquetar). Esto nos da una idea del esfuerzo necesario para mejorar los resultados etiquetando manualmente un cierto número de palabras que en apariencia resultan ser las más difíciles de resolver.

4. Resultados

Inicialmente comenzamos las pruebas con el corpus *Penn Treebank* afrontando la tarea del etiquetado POS. En la tabla 1 se muestran los resultados obtenidos en cuanto al

tamaño alcanzado y la precisión (o accuracy) con dicho corpus (seleccionamos la sección 0 de los documentos del *Wall Street Journal* como test y las secciones 1-5 y 10-19 para el corpus semilla y la fuente de frases nuevas). Tanto para el método *naive* como para el basado en *stacking*, se muestran los valores de precisión de los corpus al comienzo del experimento y al finalizar, así como la diferencia alcanzada entre estos valores. Todos estos valores se calculan para un corpus unificado generado a partir de los tres corpus que devuelve el sistema. El criterio de selección para mezclar las frases de los tres corpus se basa en las probabilidades de las etiquetas devueltas en cada caso, seleccionando la de mayor probabilidad de acierto (en el caso del método *naive* no tenemos esta probabilidad por lo que se escoge el del mejor etiquetador de entre los tres). El corpus semilla lo componen cincuenta frases y se extraen doscientas frases nuevas en cada iteración como candidatas a extender el corpus hasta alcanzar las cinco mil frases.

	Semilla	*Naive*	*Stacking*
TnT	82,27	82,43 (+0,16)	82,69 (+0,42)
TT	77,21	81,11 (+3,9)	81,4 (+4,19)
MBT	74,86	80,43 (+5,57)	81,01 (+6,15)

Tabla 1: Resultados para el corpus *Penn Treebank* (POS).

Los resultados con el etiquetado POS no parecen demasiado esperanzadores ya que la mejora lograda no resulta muy espectacular, aunque al afrontar una tarea más difícil por su mayor dependencia contextual, como es el análisis sintáctico superficial, la situación varía. Como podemos ver en la tabla 2, los resultados con el corpus del CoNLL 2000 (para la tarea de detección de sintagmas o *chunking*) muestran una caída sustancial de la precisión del sistema *naive* mientras que el sistema basado en *stacking* se muestra mucho más eficiente, siendo capaz de reponerse ante la caída inicial provocada por las primeras iteraciones del método *naive*. Esto parece indicar, que el sistema de *stacking* es capaz de responder ante la mayor dificultad de esta tarea, al poder manipular de forma más eficiente la información que subyace en el contexto de las etiquetas. Puede aprender ciertas reglas de los ejemplos que poseemos al principio del proceso para responder de forma razonable ante los casos nuevos que van apareciendo en las sucesivas iteraciones.

	Semilla	*Naive*	*Stacking*
TnT	76,5	70,18 (-6,32)	76,99 (+0,49)
TT	68,98	69,79 (+0,81)	72,71 (+3,73)
MBT	74,19	70,04 (-4,15)	76,14 (+1,95)

Tabla 2: Resultados para el corpus CoNLL 2000 (*chunking*).

La figura 3 muestra la forma en que evolucionan los corpus unificados generados por ambos sistemas, desde la semilla hasta alcanzar el objetivo de las cinco mil frases.

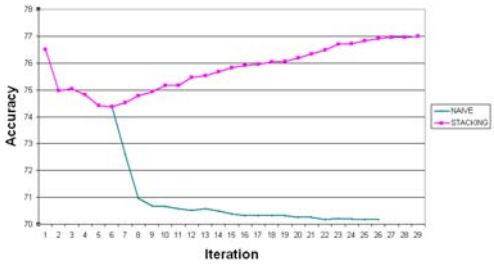

Figura 3: Evolución de los corpus unificados (evaluados con TnT).

Finalmente probamos el comportamiento de nuestro sistema introduciendo la fase de revisión en sus dos versiones (automática y mediante *active learning* simulado) tras la fase inicial de *naive*. En la tabla 3 observamos una respuesta positiva del sistema y los efectos en los resultados de etiquetar únicamente el 1,58 % de las palabras del corpus (propuestas por el sistema de revisión en los casos en los que la fiabilidad es demasiado baja).

	Semilla	Automático	Activo
TnT	76,5	78 (+1,5)	78,16 (+1,66)
TT	68,98	73,56 (+4,58)	73,89 (+4,91)
MBT	74,19	77,7 (+3,51)	77,84 (+3,65)

Tabla 3: Resultados con revisión automática y activa insertando 1129 etiquetas de 71112 (corpus CoNLL 2000).

Aunque no existe mucha diferencia entre los resultados de la revisión automática y la activa, observamos que la base de conocimiento generada es mucho más sólida en el segundo caso, quedando patente al introducir una segunda fase de revisión automática al final del experimento. Los resultados se incrementan en dos puntos respec-

to al logrado por el mejor sistema (TnT) de forma aislada con el corpus semilla inicial, mejorando los resultados obtenidos con ambas revisiones automáticas como se aprecia en la figura 4.

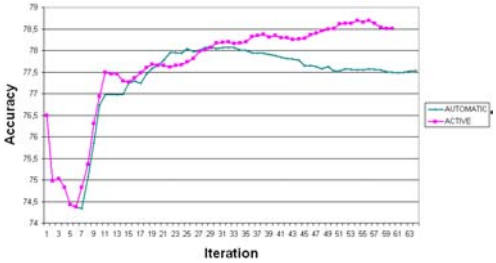

Figura 4: Evolución con dos fases de revisión (evaluando con TnT).

La tabla 4 y la figura 5 muestran la comparativa entre los valores alcanzados por el método *naive* en la tarea de *chunking* y el sistema de *stacking* con una breve fase de revisión activa aplicada sobre las cinco primeras iteraciones de *naive* finalizada con una revisión automática de todo el proceso.

	Semilla	*Naive*	*Stacking*
TnT	76,5	70,18 (-6,32)	78,51 (+2,01)
TT	68,98	69,79 (+0,81)	74,06 (+5,08)
MBT	74,19	70,04 (-4,15)	78,3 (+4,11)

Tabla 4: Comparación final entre *naive* y *stacking* (corpus CoNLL 2000).

La diferencia es obvia siempre considerando el mínimo esfuerzo empleado en este experimento para etiquetar el 1,58 % de forma manual durante la revisión activa, y el gran número de parámetros que aún debemos estudiar para detectar la configuración óptima que nos aporte los mejores resultados. Factores como el número de frases insertadas en cada iteración, el umbral para las diferentes fases o la propia configuración de las frases en el experimento pueden afectar en mayor o menor medida al resultado aun cuando las variaciones son pequeñas.

4.1. Trabajo relacionado

Hay muchos trabajos que intentan aplicar técnicas de *co-training* a tareas PLN (por ejemplo (Zavrel, 2000) y (Cucerzan, 2002)). Uno de los que más llamó nuestra atención y ha sido fuente de inspiración clara en nuestros experimentos ha sido (Clark, 2003), que

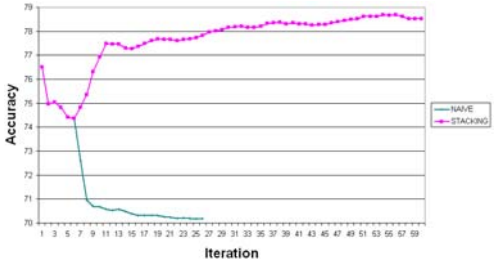

Figura 5: Comparación final entre *naive* y *stacking*.

usa la técnica del *co-training* aunque emplea un método de selección muy diferente. Está basado en acuerdos y trabaja seleccionando aleatoriamente un subconjunto de frases nuevas y calculando una tasa de acuerdo entre los diferentes etiquetadores. La figura 6 muestra su algoritmo de selección.

C es un grupo de frases etiquetadas por et_1
U es un grupo de frases sin etiquetar
Inicialización:
$c_{max} \leftarrow \emptyset$
$A_{max} \leftarrow 0$
Repetir n **veces**
  $c \leftarrow$ muestra aleatoria de C
  $et_2' \leftarrow$ etiquetador et_2 enriquecido con c
  $A \leftarrow$ acuerdo entre et_1 y et_2' sobre U
  Si $(A > A_{max})$
    $A_{max} \leftarrow A$
    $c_{max} \leftarrow c$
  Fin si
Fin repetir

Figura 6: Algoritmo de selección en (Clark, 2003) para decidir el incremento del corpus asociado al etiquetador et_2.

El método de selección seguido por (Clark, 2003) es muy costoso, aun habiendo reducido significativamente número de búsquedas al explorar sólo n subconjuntos aleatoriamente en lugar de analizar todas las posibles particiones. A pesar de este esfuerzo computacional no mejoran los resultados del método ingenuo de selección (incluirlas todas).

La principal motivación de nuestro trabajo es incluir un criterio de selección más inteligente, lo que se consigue mediante el uso de algoritmos de aprendizaje automático sobre los ejemplos producidos por los distintos etiquetadores. El coste computacional es mucho menor y los resultados que hemos obtenido en nuestras primeras pruebas son esperanzadores.

5. Conclusiones

Hemos implementado un sistema de generación automática de corpus etiquetados con el único requisito de disponer previamente de un número pequeño de frases ya etiquetadas. En el proceso, este corpus semilla será extendido hasta que se alcance el tamaño deseado del corpus final. El sistema basado en el *co-training* emplea el esquema de *stacking* para mejorar los resultados de la línea base, constituida por un esquema sencillo que hemos llamado *naive*. Durante la ejecución, el sistema se muestra estable en diferentes tareas ofreciendo mejores resultados en tareas más difíciles, en las que se requiere mayor información contextual.

Quedan muchas líneas abiertas por las que continuar en el futuro, como por ejemplo la investigación de nuevas formas de combinar las salidas de los etiquetadores para la construcción del corpus resultado, la definición de umbrales variables en las fases de *stacking* y de revisión que vayan adaptándose a la calidad del corpus disponible, o la aplicación a otro tipo de tareas más complejas que el etiquetado de palabras.

Bibliografía

S. Abney: Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. (2002) 360–367

A. Blum, T. Mitchell: Combining labelled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*. (1998) 92–100

T. Brants: TnT. A statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP00)*. (2000) 224–231

S. Clark, J. R. Curran, and M. Osborne: Bootstrapping POS taggers using Unlabelled Data. In *Proceedings of CoNLL-2003*. (2003) 49–55

S. Cucerzan, D. Yarowsky: Bootstrapping a multilingual part-of-speech tagger in one person-day. In *Proceedings of the 6th Workshop on Computational Language Learning*. (2002)

W. Daelemans, J. Zavrel, P. Berck, S. Gillis: MBT: A MemoryBased Part of Speech Tagger-Generator. In *Proceedings of the 4th Workshop on Very Large Corpora*. (1996) 14–27

Florian, R., Yarowsky, D., 2002. Modeling Consensus: Classifier Combination for Word Sense Disambiguation. In *Proceedings of EMNLP'02*, Philadelphia, pp 25–32.

Florian, R., Ittycheriah, A., Jing, H., Zhang, T., 2003. Named Entity Recognition through Classifier Combination. In *Proceedings of CoNLL-2003*, Canada, pp 168–171.

Halteren, v.H., Zavrel, J., Daelemans, W., 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics 27*, pp 199–230.

Henderson, J.C., Brill, E., 1999. Exploiting diversity in natural language processing. Combining parsers. In *1999 Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, ACL*, USA, pp 187–194.

H. Schmid: Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the Conference on New Methods in Language Processing*. (1994)

I.H. Witten, E. Frank: Data Mining. Machine Learning Algorithms in Java. Morgan Kaufmann Publishers (2000)

J. Zavrel, W. Daelemans: Bootstrapping a tagged corpus through combination of existing heterogeneous taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. (2000) 17–20