Modélisation du prétraitement des textes

Thomas Heitz[1]

[1]Université Paris-Sud, LRI Bât 490, 91405 Orsay Cedex France

heitz@lri.fr – http://www.lri.fr/~heitz/

Abstract

In this paper, we define a model for the stage of text pretreatment within the framework of text extraction and more generally information extraction from texts. This paper does not contain the details of the implementation. The purpose is to obtain a generic model of standardization of raw texts. The motivation of this paper is to generalize rather confidential and specialized works which exists for this stage of pretreatment. However we can't get away from this stage and on its quality depends largely the analysis obtained with all the later stages.

Résumé

Dans cet article, nous définissons un modèle pour l'étape de prétraitement des textes dans le cadre de la fouille de textes et plus généralement de l'extraction d'informations à partir de textes. Cet article ne contient pas les détails de l'implémentation. L'objectif est d'obtenir un modèle générique de normalisation des textes bruts. La motivation de cet article est de généraliser les travaux assez confidentiels et spécialisés qui existent pour cette étape de prétraitement. Cette étape est pourtant incontournable et d'elle dépend grandement la qualité des analyses obtenues à toutes les étapes ultérieures.

Mots-clés : fouille de textes, extraction d'informations, normalisation, prétraitement, lemmatisation, racinisation, segmentation en mots, segmentation en phrases.

1. Définition et motivation

Il existe de nombreux types de données à partir desquels des informations pertinentes pour une tâche donnée peuvent être extraites. Cependant, quel que soit le type de données, il est nécessaire de prétraiter les données brutes afin de pouvoir ensuite les traiter avec des processus unifiés et non une multitude de processus adaptés à tous les cas possibles. Ces prétraitements précèdent les traitements d'extraction d'informations même si la frontière est parfois floue. Ce type de traitements prend actuellement le nom de normalisation, ce qui reflète en partie ce en quoi ils consistent.

En effet, ces prétraitements ne consistent pas seulement à **normaliser** les valeurs des données afin d'obtenir des résultats cohérents mais aussi, dans la mesure du possible, à **corriger** certaines erreurs et à **expliciter** des informations manquantes à l'aide parfois de ressources externes.

Dans cet article, nous nous intéressons aux données textuelles qui sont parmi celles nécessitant le plus de prétraitements de par leur ambiguïté intrinsèque. En effet, la recherche de l'économie du nombre de mots et la richesse du langage pousse les locuteurs à laisser implicites de nombreuses informations.

Les prétraitements des données textuelles consistent à **normaliser** les diverses manières d'écrire un même mot, à **corriger** les fautes d'orthographe évidentes ou les incohérences

typographiques et à **expliciter** certaines informations lexicales[1] exprimées implicitement dans les textes. Les traitements pour ce dernier cas consistent, par exemple, à remplacer l'élision *l'* par les articles *le, la, les* correspondant ou à extraire la structure superficielle des textes à partir d'indices comme une ligne vide pour délimiter les paragraphes.

Notre expérience dans la participation aux compétitions TREC Novelty 2004 (Amrani et al., 2004) et TREC Genomics 2005 nous à permis d'avoir une vision concrète des nombreuses difficultés liées au prétraitement des textes. Ainsi, dans le premier cas, nous avons traité un corpus d'articles d'actualités américaines de 8 Mo avant normalisation et 6 Mo soit un million d'expressions séparées par des espaces après normalisation à l'aide de 120 règles. Dans le second cas, nous avons traité un corpus d'articles de génomique de 800 Mo avant normalisation et 500 Mo soit 80 millions d'expressions séparées par des espaces après normalisation à l'aide de 150 règles. Dans les deux cas, les traitements durent moins d'une minute par méga-octet.

Dans la suite de cet article, nous commençons par établir un état de l'art succinct du prétraitement des textes étant donné que le sujet a donné lieu à peu de publications. Puis, nous présentons quelles sont les incohérences et ambiguïtés à traiter dans les textes bruts. Ensuite, nous proposons un modèle de dépendances de la chaîne de traitements et étudions les cas critiques. Enfin, nous concluons sur quelques perspectives de développements futurs.

2. État de l'art

Est-il possible de se passer de normalisation ? Bien que la plupart des articles en extraction d'informations passent cette étape sous silence, il semble malheureusement impossible d'obtenir des résultats pertinents à partir de corpus non normalisés car il n'existe pas d'outils de fouille de textes résistants à un bruit trop important. Tout au plus, des approches essaient de se passer de certaines parties de la normalisation comme (Dias et al., 2000) qui se base uniquement sur des statistiques de mots sans lemmatisation ni groupement des locutions. De plus, toute l'information présente dans un corpus n'est pas nécessairement utile pour une tâche donnée et il faudra sélectionner la partie adéquate. De même, pour lever certaines ambiguïtés ou réduire le nombre d'opérations effectuées, il est souvent nécessaire d'orienter les traitements suivant l'objectif final qui peut être aussi différent que la traduction et l'indexation de textes.

Deux principaux prétraitements sont présents dans la littérature du domaine de la fouille de textes. D'une part, les méthodes de lemmatisation des formes fléchies des mots et d'autre part les méthodes de segmentation d'un texte, notamment en phrases.

La **lemmatisation** (Korenius et al., 2004) qui consiste à réduire les mots à leur lemmes, au masculin singulier ou à l'infinitif, ou la racinisation (stemming) (Tomlinson, 2004 ; Hull et Grefenstette, 1996) qui consiste à réduire les mots à leur radicaux après effacement des affixes sont des techniques de normalisation déjà bien étudiées et reposent sur des grammaires et des lexiques actuellement assez complets même si le traitement des mots inconnus reste un problème pour la lemmatisation.

Les méthodes de **segmentation** des textes par apprentissage des frontières des phrases (Zhang et al., 2003) ou par utilisation d'automates et de listes élaborés manuellement (Friburger et al.,

[1] informations supprimées pour des raisons d'économie du langage comme les élisions, les abréviations, les marques de fin de phrase, etc.

2000) font aussi l'objet d'un nombre significatifs d'articles bien que chaque méthode reste très spécifique au corpus traité.

Pour le reste des prétraitements, notamment ceux qui précèdent la lemmatisation et la segmentation, il n'existe que peu d'articles (Grefenstette et Tapanainen, 1994 ; Habert et al., 1998 ; Mikheev, 2000 ; Sagot et Boullier, 2005) traitant explicitement du prétraitement des données textuelles. Les traitements proposés dans ces articles plus ceux d'autres outils existants sont résumés dans le tableau 1.

Description du traitement	Étape	Description du traitement	Étape
1. (Grefenstette et Tapanainen, 1994)		**4. Gate 3.0 (Cunningham et al., 2002)**	
1. Supprimer les étiquettes SGML	I2	1. Segmentation en mots	E2
2. Recoller les césures	I3	2. Lemmatisation	E2
3. Marquer les nombres et les abréviations	A1	3. Identification des entités nommées	A1
4. Segmenter le texte en phrases	E1	4. Découpage en phrases	E1
2. Multext multilingual segmenter tools (v 1.3.1),		**5. SXPIPE 1.0 (Sagot et Boullier, 2005)**	
1. Séparer le texte selon les espaces	A2	1. Identification des courriels, dates, adresses, etc.	A1
2. Isoler les ponctuations	A1	2. Détection des frontières des phrases	A2
3. Fusionner les ponctuations composées	A1	3. Identification des mots inconnus	A1
4. Séparer les expressions avec ponctuations	A1	4. Identification des acronymes, noms propres	A1
5. Fusionner les abréviations composées	A1	5. Identification des mots étrangers	A1
6. Identifier les abréviations	A1	6. Découpage en mots et correction orth.	E2/I3
7. Recombiner les unités multimots	E2	7. Identification des nombres en lettres	A1
8. Identifier les dates	A1	8. Identification des mots composés	E2
9. Identifier les nombres	A1	9. Réaccentuation et recapitalisation	I3
10. Identifier les énumérations	A2		
11. Détecter les frontières des phrases	E2		
3. (Adda et al., 1997)		**6. Notre modèle (Amrani et al., 2004)**	
0. Encodage des accents et autres diacritiques	I1	1. Remplace les caractères non ASCII et entités	I1
0. Prétraitement des nombres et unités	I3	2. Convertit le document en format texte linéaire	I2
0. Correction du formatage et des ponctuations	I3	3. Met un paragraphe par ligne	A2
0. Traitement des ponctuations non ambiguës	I3	4. Normalise le paragraphe - Incohérences	I3
0. Séparation en articles, paragraphes, phrases	E2	5. Normalise le paragraphe - Ambiguïtés	A1
1. Traitement des ponctuations ambiguës	A1	6. Recherche des frontières des phrases	A1, A2
2. Traitement des débuts de phrase capitalisés	A1	7. Met une phrase par ligne	E1
3. Traitement des nombres	A1	8. Normalise les phrases	E2
4. Traitement des acronymes	A1		
5. Traitement des capitales emphatiques	A1	**7. (Mikheev, 2000)**	
6. Décomposition	E2	1. Classer les expressions terminées par un point	A1
7. Pas de distinction de casse	I3	2. Classer les mots capitalisés après un point	A1
8. Pas de diacritiques	I1	3. Assigner les fins de phrases	E1

Tab. 1 - *Enchaînements des traitements proposés dans différentes chaînes de prétraitement des textes. Les étapes sont définies dans la section 3.*

Les procédures d'analyse des étapes ultérieures au prétraitement comme l'étiquetage et la terminologie se sont affinées et sont donc de plus en plus sensibles à la qualité des données initiales. L'utilisation de corpus de textes normalisés principalement à la main devient de plus

en plus un frein à l'utilisation de données réelles et diversifiées. C'est pourquoi cet article se veut une première réponse à la définition d'un modèle du prétraitement des textes.

Après ce court état de l'art, nous nous interrogeons sur les incohérences et ambiguïtés à traiter dans les textes bruts puis sur les différents types de normalisations.

3. Incohérences, ambiguïtés et expressions à traiter

Il existe deux grands types de difficultés à traiter, en premier lieu, dans les textes bruts : les incohérences et les ambiguïtés. Celles-ci doivent être traitées avant toute extraction d'informations.

Les incohérences (noté I1, I2 et I3 dans les tableaux 1 et 2) doivent être **corrigées** afin de permettre un traitement unifié lors des étapes ultérieures d'extraction d'informations.

Le **format variable d'encodage** des textes, p. ex. des caractères accentués encodés de manière différente d'un texte à l'autre, constitue un premier type d'incohérences. Ce type d'incohérence est à traiter en premier lieu car le problème se situe au niveau du caractère et il faut décider d'un encodage unique avant de pouvoir effectuer des traitements au niveau des mots (noté I1 dans les tableaux 1 et 2).

Les **marqueurs incohérents** de présentation ou de sens, p. ex. des balises XML incohérentes d'un texte à l'autre, constituent un second type d'incohérences. Il s'agit de supprimer ou de transformer ces marqueurs car ils sont souvent incohérents d'un standard à l'autre, voire d'une version à l'autre. Le but est donc de trouver une partie commune qui soit exploitable pour la suite des traitements (I2).

La présence de **fautes d'orthographe** ou d'**incohérences typographiques**, p. ex. des majuscules incorrectement utilisées, constitue un troisième type d'incohérences. Ce traitement devrait être effectué avant celui des ambiguïtés lexicales pour limiter les erreurs dues aux données erronées (I3).

Les ambiguïtés (A1 et A2) doivent être **explicitées**, dans la mesure du possible, afin que les traitements ultérieurs de niveau syntaxique ou sémantique ne soient pas trop pénalisés par le manque d'informations des niveaux inférieurs.

Les **ambiguïtés lexicales** nécessitant parfois des ressources externes aux textes afin de les lever, p. ex. un lexique d'abréviations, constituent un premier type d'ambiguïtés. Ce traitement devrait être effectué avant la celui de la structure des textes pour ne plus avoir le problème des abréviations en fin de phrase lors de la recherche des fins de phrase (A1).

La **structure des textes** non encodée ou de façon partielle, p. ex. la fin des phrases et des paragraphes, constitue un second type d'ambiguïtés. Il s'agit de marquer les frontières à l'aide de balises (A2).

Une fois les incohérences corrigées et les ambiguïtés explicitées, il devient possible de **normaliser** certaines expressions (E1 et E2) des textes. Différents types de normalisations existent et débordent souvent du cadre des prétraitements.

La **segmentation** finale en phrases est rendue possible grâce aux informations recueillies lors du traitement des ambiguïtés A1 et A2 ainsi que par l'utilisation de règles supplémentaires lorsque les informations déjà connues ne permettent pas de décider immédiatement (E1).

La **lemmatisation**, la racinisation, le groupement des locutions et la segmentation finale en mots viennent logiquement en dernier lorsque les incohérences et les ambiguïtés ont été traitées et qu'enfin les phrases ont été segmentées (E2).

Dans la section suivante, nous établissons les différentes dépendances entre chaque traitement et discutons des cycles existants.

4. Dépendances entre les traitements

Dans le tableau 2, les dépendances sont écrites sous la forme de couples de dépendance séparés par des virgules : *étape1 étape_suivant_l'étape1, étape2 étape_suivant_l'étape2, etc.* Nous effectuons un tri topologique sur les dépendances et nous donnons les principaux cycles existants. Si les cycles sont justifiés alors nous donnons une condition d'arrêt pour empêcher les boucles infinies de calcul.

Dépendances théoriques décrites dans la section 3	
Dépendances	I1 I2, I2 I3, I3 A1, A1 A2, A2 E1, E1 E2
Cycles	pas de cycle
Dépendances réelles décrites dans le tableau 1	
Dépendances 1	I2 I3, I3 A1, A1 E1
Cycles	pas de cycle
Dépendances 2	A2 A1, A1 E2, E2 A1, A1 A2, A2 E2
Cycles	cycle A1 A2 ; cycle A1 E2
Dépendances 3	I1 I3, I3 E2, E2 A1, A1 E2, E2 I3, I3 I1
Cycles	cycle I3 E2 ; cycle I1 I3 ; cycle A1 E2
Dépendances 4	E2 A1, A1 E1
Cycles	pas de cycle
Dépendances 5	A1 A2, A2 A1, A1 E2, A1 I3, E2 I3, E2 A1, I3 A1, A1 E2, E2 I3
Cycles	cycle A1 A2 ; cycle A1 E2 ; cycle A1 I3
Dépendances 6	I1 I2, I2 A2, A2 I3, I3 A1, A1 A2, A2 E1, E1 E2 A1 E2
Cycles	cycle A1 A2 I3
Dépendances 7	A1 E1
Cycles	pas de cycle

Tab. 2 - *Couples de dépendances séparés par des virgules et cycles entre les traitements effectués lors du prétraitement des textes.*

Les cycles existants entre des étapes appartenant au même ensemble, c'est-à-dire I1, I2, I3 ou A1, A2 ou E1, E2, E3, posent moins de problèmes que les cycles entre ensembles différents. C'est pourquoi nous n'étudierons que ces derniers. De plus, les enchaînements problématiques sont ceux qui entraînent un retour vers un traitement de plus bas niveau, c'est-à-dire E A, E I ou A I.

Un **premier cycle** entre ensembles différents dans les dépendances 2 est : A1 E2. L'enchaînement de retour E2 A1 signifie que les unités multimots (E2) sont recombinées avant l'identification des dates (A1). Or, le groupement des locutions devrait être effectué après l'identification des expressions ambiguës telles les nombres, dates et abréviations. Car, même si le classement de ces entités nommées est sommaire, il permet d'éviter de regrouper dans une locution un mot commun et une entité nommée. Ce cycle n'est donc pas justifié.

Un **second cycle** dans les dépendances 3 est : I3 E2. L'enchaînement de retour E2 I3 signifie que la décomposition en mots (E2) est effectuée avant la correction d'incohérences typographiques (I3). En fait, cela est un cas un peu particulier à la chaîne de traitement (Adda et al., 1997) qui transforme les majuscules en minuscules à la fin pour effectuer des mesures de couverture du lexique. Seulement, il y a une perte d'informations pour les traitements ultérieurs qui voudraient s'appuyer sur la casse des mots. Il faudrait donc supprimer cette étape si la chaîne de prétraitements devait s'intégrer dans une chaîne globale de fouille de textes. Ce cycle n'est justifié que pour les besoins de l'évaluation.

Un **troisième cycle** dans les dépendances 5 est : A1 I3. L'enchaînement de retour A1 I3 signifie que l'identification des mots inconnus, noms propres et mots étrangers (A1) est effectuée avant la correction orthographique (I3). En fait, il y a bien une pré-correction orthographique lors des identifications, elle n'est pas effectuée mais sert à trouver les mots qui ne sont pas reconnus par un lexique ou facilement corrigeables. Par contre, il est intéressant de noter qu'une correction orthographique, effective celle-ci, sera effectuée plus tard simultanément au découpage en mots (E2). On peut donc noter qu'un traitement peut être effectué plusieurs fois de façon incrémentale afin de n'utiliser qu'une partie des résultats dans un premier temps. Il n'y a donc finalement pas de cycle.

Un **quatrième cycle** dans les dépendances 6 est : A1 A2 I3. L'enchaînement de retour A2 I3 signifie que le texte est segmenté en paragraphes (A2) puis que les incohérences typographiques (I3) sont corrigées. Or, les séparateurs de paragraphes peuvent être incohérents. Ce cycle n'est donc pas justifié.

Il est difficile de trouver de vrais cycles au niveau de détails que nous avons utilisé précédemment. Cependant, cela nous a permis de mettre en évidence des enchaînements de retour non justifiés. Voici maintenant quelques exemples, à un niveau plus détaillé, de cycles que nous avons rencontré lors de nos expérimentations durant des compétitions de fouille de textes.

Sur le **corpus d'articles d'actualités** américaines de TREC Novelty (Amrani et al., 2004), nous avons obtenu un cycle entre la correction des élisions en anglais (A1) et le groupement des locutions (E2). En effet, si les élisions du verbe être *'s*, par exemple, sont remplacées par *is* dans le texte, alors le lexique des locutions doit contenir *what_is_more* comme adverbe tandis que *what_'s_more* sera inutile et inversement si le lexique contient uniquement les locutions avec les élisions de *is* telle *what_'s_more* alors il ne faut pas remplacer les élisions auparavant. La condition d'arrêt est satisfaite lorsque la correction ou pas des élisions de *is* est cohérente avec le lexique des locutions. Au moins deux solutions existent : mettre les deux formes dans le lexique ou appliquer les mêmes remplacements dans le texte et dans le lexique. Ce problème se généralise à tous les remplacements lors du traitement des ambiguïtés lexicales (A1) telles les élisions, les abréviations, les contractions, etc.

Sur le **corpus d'articles de génomique** de TREC Genomics, nous avons obtenu un cycle entre la recherche des frontières des phrases (A1) et la segmentation en phrases (E1). En effet, à chaque modification de la définition des frontières des phrases, la segmentation en phrases est modifiée et inversement si de nouvelles erreurs apparaissent ou qu'il reste de nombreux cas non traités dans la segmentation en phrases alors la définition des frontières doit être revue. L'utilisateur arrête le cycle lorsque le traitement devient trop lent, lorsque le temps nécessaire à améliorer la définition des frontières est excessif par rapport au nombre de cas erronés restant pouvant être traités ou encore lorsque toute modification de la définition rajoute plus d'erreurs qu'elle n'en corrige.

Après avoir illustré notre modèle sur différents exemples, nous allons donner quelques perspectives et conclure.

5. Perspectives

Il serait intéressant d'effectuer des expérimentations sur différents enchaînements des traitements et d'évaluer la qualité des résultats obtenus pour des objectifs et des domaines précis comme la recherche de termes dans des textes de génomique ou encore des systèmes de question-réponse sur des textes journalistiques, etc. Ainsi, nous pourrions montrer quels types d'enchaînements de prétraitements se prêtent le mieux à tel domaine et objectif final.

Par contre, la comparaison de résultats d'expérimentations sur un même corpus de textes et une même tâche nous semble peu pertinente puisque les résultats restent uniquement valables pour ce cas particulier.

Nous n'avons pas donné de résultats d'évaluations comparatives dans cet article car nous avons préféré donner plus de place à la présentation d'un modèle élaboré à l'aide des recherches déjà effectuées et de nos propres expérimentations. Cet article n'en demeure pas moins basé sur plusieurs mois d'expérimentations pour au moins deux compétitions de fouille de textes, TREC 2004 et 2005 (Amrani et al., 2004).

Dans le futur, l'élaboration d'une interface graphique simple avec une visualisation rapide des résultats permettrait de tester ce modèle avec des utilisateurs non informaticiens. Il est possible notamment d'utiliser des cadres de développement pour la fouille de textes (Cunningham et al., 2002) qui permettent de facilement combiner des traitements, de les tester sur des textes et comparer les résultats entre deux enchaînements.

Enfin, le format de sortie des résultats pourrait utiliser un des premiers standards d'encodage pour les annotations morpho-syntaxiques (Clément et Villemonte de la Clergerie, 2005).

6. Conclusion

Dans cet article, nous avons donné une définition, que nous espérons complète, du prétraitement des textes, ce qui n'a pas déjà été fait à notre connaissance. De plus, un état de l'art relativement complet a été fait et surtout, pour chaque article, une synthèse des différents traitements avec leurs enchaînements.

Puis, nous avons cherché à théoriser quelque peu cette étape de base de la fouille de textes. Pour cela, nous avons introduit l'idée de cycles et de conditions d'arrêt qui nous semblent des outils efficaces pour décrire les systèmes de prétraitement des textes et plus généralement les chaînes de fouille de textes.

Enfin, plusieurs exemples venant en partie de nos propres expériences ont permis d'illustrer ce modèle.

Je remercie Yves Kodratoff pour m'avoir aidé à éclaircir certains passages.

Références

Adda, G., M. Adda-Decker, J. Gauvain, et L. Lamel. (1997). Text normalization and speech recognition in french. In Proceedings of the European Conference on Speech Technology, EuroSpeech, Rhodes, Volume 5, pp. 2711-2714.

Amrani, A., J. Azé, T. Heitz, K. Kodratoff, et M. Roche (2004). From the texts to the concepts they contain : a chain of linguistic treatments. In Proceedings of TREC'04 (Text REtrieval Conference), National Institute of Standards and Technology, Gaithersburg Maryland USA, pp. 712-722.

Sagot, B. et P. Boullier (2005). From raw corpus to word lattices : robust pre-parsing processing [d'un corpus brut à des treillis de mots : traitement pré-syntaxique robuste]. In Proceedings of L&T 05, Poznan, Pologne, pp. 348-351.

Clément, L. et E. Villemonte de la Clergerie (2005). MAF : a morphosyntactic annotation framework. In proc. of the 2nd Language & Technology Conference (LT'05), Poznan, Poland, pp. 90-94.

Cunningham, H., D. Maynard, K. Bontcheva, et V. Tablan (2002). GATE : A framework and graphical development environment for robust NLP tools and applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics.

Dias, G., S. Guilloré, et J. P. Lopes (2000). Extraction automatique d'associations textuelles à partir de corpora non traités. In M. Rajman and J. Chapelier (Eds.), Actes des 5es Journées Internationales d Analyse Statistique des Données Textuelles (JADT 2000), Volume 1, pp. 213-221.

Friburger, N., A. Dister, et D. Maurel (2000). Améliorer la reconnaissance automatique des fins de phrases. In Actes des troisièmes journées Intex, Liège, 13-14 juin 2000 (Anne Dister Ed.), dans Revue, Informatique et Statistiques dans les sciences humaines 36 numéo 1-4, pp. 181-200.

Grefenstette, G. et P. Tapanainen (1994). What is a word, what is a sentence ? problems of tokenization. In Proceedings of the 3rd International Conference on Computational Lexicography, Budapest, pp. 79-87.

Habert, B., G. Adda, M. Adda-Decker, P. Boula de Mareuil, S. Ferrari, O. Ferret, G. Illouz, et P. Paroubek (1998). Towards tokenization evaluation. In Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain, pp. 427-431.

Hull, D. et G. Grefenstette (1996). Stemming algorithms: A case study for detailed evaluation. Journal of the American Society for Information Science 47(1), 70-84.

Korenius, T., J. Laurikkala, K. Jarvelin, et M. Juhola (2004). Stemming and lemmatization in the clustering of finnish text documents. In CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management, New York, NY, USA, pp. 625-633. ACM Press.

Mikheev, A. (2000). Document centered approach for text normalization. In Proceedings of the 23rd ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, pp. 136-143.

Tomlinson, S. (2004). Lexical and algorithmic stemming compared for 9 european languages with hummingbird searchserver. In CLEF 2003, Lecture Notes in Computer Science, Volume 3237, pp. 286-300.

Zhang, T., F. Damerau, et D. Johnson (2003). Updating an nlp system to fit new domains: an empirical study on the sentence segmentation problem. In W. Daelemans et M. Osborne (Eds.), Proceedings of CoNLL-2003, pp. 56-62. Edmonton, Canada.