Utilisation des poids empiriques dans l'analyse syntaxique : une application en Traduction Automatique

L. L. Iomdin, V. G. Sizov et L. L. Tsinman

Traduit du russe par Jasmina Milićević

Abstract. Empirical Weights in Parsing. The paper discusses a complex of solutions aimed at ambiguity resolution in the parsing component of a multipurpose NLP system, ETAP-3. The main idea is to introduce a system of priorities, or weights, dynamically produced for the elements of the text processed and of the structure generated during all parsing phases. These weights, empirically assigned to lexical entries and fragments of parsing rules, help tune the parser to the generation of an optimal syntactic structure of an ambiguous sentence.

1. Introduction

Le problème de résolution des ambiguïtés sémantiques et syntaxiques dans le contexte du traitement automatique de la langue existe depuis aussi longtemps que ce domaine lui-même, mais, jusqu'à présent, aucune solution générale satisfaisante à ce problème n'a été proposée. Il est possible qu'une telle solution ne puisse pas être trouvée du tout, étant donné que le problème en question est un des plus complexes, sinon le plus complexe, parmi ceux qui se posent pour un analyseur de texte. Une résolution efficace de l'ambiguïté dans le processus de l'analyse présuppose l'accès non seulement à des données sémantiques complexes, mais aussi à des informations variées sur l'état du monde (qui ne se prêtent pas bien à une représentation formelle). Cet état des choses a un caractère universel et ne dépend ni du modèle linguistique sous-jacent à l'analyseur ni du type d'application dans lequel celui-ci est utilisé (Oepen *et al*. 2000, 3*ff*). Ce que l'homme fait facilement, en s'appuyant dans son choix de l'interprétation des éléments ambigus du texte sur le sens commun, sur ses connaissances du monde et sur le contexte de communication plus large, n'est pas encore possible pour les systèmes informatiques.

Cependant, une solution partielle à ce problème est tout à fait possible et tout système d'analyse de la langue naturelle dispose d'un ensemble de mécanismes plus ou moins efficaces visant la résolution de l'ambiguïté. À notre avis, cette tâche ne peut être menée à bien que lorsque les mécanismes correspondants sont bien fondés du point de vue linguistique.

Dans ce qui suit, nous présentons un ensemble de solutions utilisées pour la résolution des ambiguïtés dans le système polyvalent de traitement automatique de la langue, ETAP-3 (Apresjan *et al*. 1989, Iomdin et Cinman 1997, Boguslavskij *et al*. 2000). Ces solutions se basent sur un jeu de priorités qui sont dynamiquement assignées aux éléments du texte analysé et aux structures générées à des étapes différentes de l'analyse. Les priorités, qu'ETAP-3 utilise dans le cas d'ambiguïté du texte de départ pour en construire la structure optimale, sont calculées grâce à un mécanisme élaboré de poids (angl. *weights*) que les linguistes assignent de façon empirique aux unités lexicales et règles syntaxiques, au moyen d'instructions spéciales.

2 Formulation de la tâche

Comme on le sait bien, toute langue naturelle est ambiguë de par sa nature. Tout texte, indépendamment de son genre, caractère, thème ou volume, contient inévitablement des éléments qui correspondent à plus d'un sens. Dans un texte écrit, peuvent être ambigus :

- Mots-formes (la forme *vesla* peut être le génitif singulier ou l'accusatif pluriel du lexème VESLO 'aviron') – dans des cas comme celui-ci il s'agit de l'**homonymie morphologique**.

- Lexèmes (MIR1 ‘univers’ vs. MIR2 ‘absence de guerre’, UČENIE1 ‘doctrine’ vs. UČENIE 2 ‘entraînement’, UČIT´1 ‘enseigner’ vs. UČIT´2 ‘apprendre’) – dans des cas comme celui-ci on parle de l’**homonymie lexicale** ou de **polysémie.**
- Constructions (*Rossijskoj sbornoj nel´zja proigrat´* ≈ lit.‘Il n’est pas possible à l’équipe nationale russe de perdre’ *vs*. lit.‘Il n’est pas possible de perdre à l’équipe nationale russe.’) – il s’agit des cas de l’**homonymie syntaxique**.

Il arrive fréquemment que des différents types d’ambiguïté sont présents simultanément dans un texte. Par exemple, le mot-forme *stekla* peut appartenir au lexème nominal STEKLO ‘verre’ et au lexème verbal STEKAT´ ‘dégouliner’ ; le mot-forme *stekli* peut appartenir aux deux lexèmes verbaux différents — STEKAT´ ‘dégouliner’ et STEKLIT´ ‘vitrer’. Dans de pareils cas, on parle de l’**homonymie lexico-morphologique** et **lexico-grammaticale**.

Dans la phrase (1) *VVS ožidajut sokraščenia*, qui peut être interprétée soit comme (1a) lit.‘Les force aériennes attendent des réductions’, soit comme (1b) lit.‘Des réductions attendent les forces aériennes’, on est en présence de (i) l’homonymie morphologique des mots-formes VVS (le nominatif en (1a) *vs*. l’accusatif en (1b)) et *sokraščenia* (l’accusatif en (1a) et le nominatif en (1b)) ; (ii) l’homonymie syntaxique (les mots-formes *VVS* et *sokraščenia* fonctionnent comme le sujet ou comme l’objet) et (iii) l’homonymie lexicale des verbes OŽIDAT´1 ‘être dans l‘état d’attente’ et OŽIDAT´2 ‘être probable (de se produire) dans un proche avenir’. Il s’agit, dans des cas comme celui-ci, de l’**homonymie lexico-syntaxique**.

De façon générale, la tâche de résolution des ambiguïtés dans un système d’analyse syntaxique peut être formulée comme suit : parmi toutes les interprétations possibles des éléments d’un texte, le système doit choisir celle(s) qu’aurait choisie(s) un humain en train d’analyser ce texte. Conformément à cela, plus le système s’approche de cet idéal, plus haut doit être le niveau auquel la résolution des ambiguïtés a lieu. Ceci veut dire en particulier que le système d’analyse automatique doit modéliser autant que possible sinon le processus d’analyse du texte par l’humain comme tel, au moins son résultat : si les auteurs du système ne sont pas en mesure d’utiliser, en vue de résoudre des ambiguïtés, des connaissances sur le monde ou des informations sémantiques complexes, ils doivent proposer certains mécanismes de compensation qui permettraient d’atteindre cet objectif. C’est exactement cette démarche-là qui a été adoptée dans l’ETAP-3.

3. Le jeu des priorités dans l’ETAP-3

3.1 Brève caractérisation de l’ETAP-3

Le système polyvalent de traitement automatique de la langue ETAP-3 contient plusieurs composantes majeures, dont le système de TA anglais/russe et vice-versa, le système de communication en langue naturelle avec les bases de données, le système de paraphrasage, le système de traduction du Langage de Réseaux Universel (UNL) en langue naturelle, le système de correction syntaxique du texte russe, etc. Bien que les solutions introduites pour la résolution des ambiguïtés soient utilisées par toutes les composantes du système, dans ce qui suit, pour fixer les idées, nous ne considérons que le système de TA russe/anglais. Conformément à cela, toutes les solutions seront illustrées à partir du russe, la langue-source du système.

Comme toutes les composantes d’ETAP-3 fonctionnent en traitant une phrase à la fois, lors d’une analyse, le système ne peut pas aller au-delà d’une phrase individuelle. Par conséquent, dans la résolution des ambiguïtés le système ne peut pas utiliser le contexte précédent, y compris dans des cas où les ambiguïtés dans des phrases précédentes auront dèjà été résolues avec succès. De cette façon, parlant de l’analyse du texte par l’humain, nous devons nous limiter aux situations où l’humain analyse des phrases isolées.

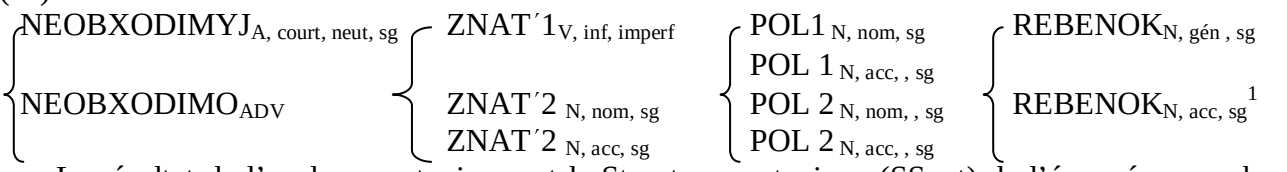
Le processus d’analyse de l’énoncé comporte les deux étapes principales suivantes :

(1) analyse morphologique (AnalyseMorph) et (2) analyse syntaxique (AnalyseSynt).

Le résultat de l’AnalyseMorph est la Structure Morphologique (SMorph) de l’énoncé, c-à-d, une suite linéaire des SMorph de tous les mots-formes constituant l’énoncé, la SMoph d’un mot-

forme étant le nom de lexème auquel appartient le mot-forme, muni des valeurs des caractéristiques flexonnelles. Si le mot-forme est ambigu, il a plusieurs SMorph. Par exemple, la SMorph de l'énoncé (2) *Neobhxodimo znat pol rebenka* se présente comme suit :

(2a)



Le résultat de l'analyse syntaxique est la Structure syntaxique (SSynt) de l'énoncé − un arbre de dépendances dont les nœuds sont étiquetés de lexèmes de l'énoncé et dont les branches portent des noms de relations syntaxiques. Par exemple, la SSynt de l'énoncé (2) se présente comme suit :

(2b) NEOBHODIMYJ$_{A,\ court,\ neut,\ sg}$ —**prédicative**→ ZNAT´1$_{V,\ inf,\ imperf}$ —**1ère complétive**→ POL1 $_{N,\ acc,\ sg}$ —**1ère complétive**→ REBENOK$_{N,\ gén,\ sg}$

lit. '[Il est] nécessaire [de] savoir [le] sexe [de l'] enfant'.

Comme le montre (2a), aucune résolution de l'ambiguïté n'est faite lors de l'analyse morphologique ; ceci veut dire que toutes les opérations visant la résolution des ambiguïtés sont effectuées lors de l'étape de l'analyse syntaxique : l'arbre de dépendance qui en résulte est constitué d'objets complètement désambiguïsés (cf. 2b). Il découle de cela que toute énoncé ambigu correspond à plus d'une SSynt. Par conséquent, un des objectifs les plus importants de l'AnalyseSynt est d'organiser la construction de l'ensemble de SSynt correspondant à l'énoncé de façon à ce que les SSynt les plus adéquates soient construites en premier.

L'étape de l'AnalyseSynt, qui prend à l'entrée la SMorph de la phrase, comporte plusieurs sous-étapes :

1) Analyse pré-syntaxique, pendant laquelle certaines opérations auxiliaires sont effectuées ;
2) Formulation des hypothèses syntaxiques – des arbres minimaux (deux nœuds reliés par une relation syntaxique) de l'arbre de dépendances à construire ; on obtient ces hypothèses en appliquant toutes les conditions sur l'ordre linéaire contenues dans l'ensemble des règles syntaxiques.
3) Sélection du sommet de l'arbre de dépendances à construire ;
4) Construction de l'arbre de dépendances – essentiellement, le choix des relations syntaxiques correctes à partir de l'ensemble des hypothèses, effectué à l'aide de divers filtres.

La résolution des ambiguïtés se fait à chacune des quatre sous-étapes de l'AnalyseSynt.

3.2. Principes généraux

Lors de l'AnalyseSynt, ETAP-3 manipule les objets de deux types – des mots-formes et des hypothèses syntaxiques (les relations syntaxiques futures, dont les noms doivent figurer dans l'arbre SSynt). Lorsqu'il apparaît pour la première fois dans le texte, chacun de ces objets a par défaut la priorité normale (par convention, 0). Pendant l'analyse, cette priorité peut augmenter ou diminuer. Cela se fait par l'application de règles spéciales, comportant des instructions, qui augmentent ou réduisent d'un point la priorité courante de l'objet. Comme pendant les différentes sous-étapes de l'analyse un même objet peut être la cible de plusieurs règles syntaxiques, sa priorité générale peut changer de plusieurs points (dans la pratique, la priorité varie entre –3 et +3, les valeurs extrêmes étant très rares de sorte que, dans la grande majorité des cas, la priorité se situe entre –1 et +1).

À chaque sous-étape de l'analyse, l'algorithme lit les priorités courantes des objets en rejetant l'un après l'autre ceux qui ont une moindre priorité. Supposons, par exemple, qu'au moment où débute la sous-étape 4 (choix des hypothèses correctes pour la construction de l'arbre) un groupe

[1] NEOBXODIMYJ $_{[ADJ]\ forme\ courte}$ 'nécessaire' ; NEOBXODIMO$_{[ADV]}$ 'nécessairement' ; ZNAT'1 'savoir' ; ZNAT'2$_{[N,\ fém]}$ 'noblesse' ; POL1$_{[N,\ masc]}$ 'sexe' ; POL2$_{[N,\ masc]}$ 'plancher' ; REBENOK$_{[N,\ neut]}$ 'enfant'.

d'hypothèses A a la priorité –2, un autre groupe d'hypothèses B a la priorité –1 et tous les autres (groupe C) a une priorité plus grande. Dans un cas comme celui-ci, l'algorithme commence par éliminer toutes les hypothèses du groupe A et essaie de construire l'arbre à partir des hypothèses restantes. Si après l'application de tous les filtres durant la sous-étape 4 le nombre d'hypothèses restantes demeure trop élevé pour qu'on puisse construire l'arbre de façon univoque, l'algorithme élimine toutes les hypothèses du groupe B et essaie de construire l'arbre de dépendances seulement à partir des hypothèses appartenant au groupe C.

Si à un pas quelconque de l'analyse il s'avère impossible de construire l'arbre, l'algorithme fait, pour ainsi dire, un pas en arrière, reprend quelques-unes des hypothèses éliminées, en commençant par celles ayant une priorité plus élevée, et y applique de nouveau les filtres. À toutes les étapes de l'AnalyseSynt, le principe général suivant est à l'œuvre : plus tôt un élément structural ayant une priorité basse a été exclu de la considération, plus tard il est réintroduit si l'algorithme doit revenir en arrière.

3.3. Exemples d'utilisation du jeu de priorités

Bien que le système des priorités à utiliser lors de l'analyse du texte ait été complètement intégré à ETAP-3 il y a à peine quelques mois, son application s'est avérée extrêmement efficace. Nous donnons ci-dessous quelques exemples typiques de son utilisation.

3.3.1. Renforcement/affaiblissement des homonymes morphologiques et syntaxiques lors de l'analyse pré-syntaxique

Des difficultés inhérentes à l'analyse de textes sont dues au fait que certaines lexies très fréquentes ont des homonymes. Ces difficultés deviennent plus grandes lorsque les mots-formes homonymes appartiennent à des lexies des classes syntaxiques différentes. Sont illustratifs à cet égard les cas d'homonymie des mots-formes comme *dlja* (préposition 'pour' *vs.* gérondif du verbe *dlit'* **poét.** 'prolonger'), *pri* (préposition 'auprès' *vs.* impératif du verbe *peret'* **vulg**. 'pousser'), *tri* и *pjat'* (numéraaux *trois* et *cinq* vs. impératifs des verbes *teret'* 'frotter' и *pjatit'* 'faire reculer'), *na* (préposition 'sur' vs. interjection 'prend') etc. Si toutes les interprétations des tels mots-formes sont considérées comme ayant la même priorité, lors de l'analyse, on risque d'obtenir en premier des SSynt exotiques, qui ne sont pratiquement jamais prises en considération par un humain analysant le texte. Pour éviter cela, il suffit de baisser la priorité du second élément de chaque paire, déjà lors de la première sous-étape de l'AnalyseSynt. Techniquement, cela se fait comme suit : dans l'article de dictionnaire du lexème dont le paradigme contient l'homonyme « rare », on inscrit une règle qui vérifie s'il coïncide avec le l'homonyme « fréquent » faisant partie du paradigme du premier élément de la paire et qui assigne à l'homonyme rare une plus basse priorité. Le retour à un homonyme rare, s'il y a lieu, se fait lors d'une phase assez avancée de l'analyse.

On a recours à une solution similaire là où le paradigme d'un lexème contient des mots-formes qui ont une basse probabilité d'occurrence. À titre d'exemple, les lexies OTEČESTVO 'patrie' et EDINSTVO 'unité' ne s'utilisent pratiquement pas au pluriel. Pour les lexies de ce type (dont une langue donnée compte des milliers), les mots-formes rares sont marqués dans le dictionnaire comme ayant une basse priorité, ce qui réduit de façon considérable le nombre des objets-candidats pour l'apparition dans la SSynt.

Finalement, à l'étape de l'analyse pré-syntaxique, on utilise souvent les règles qui renforcent ou affaiblissent des homonymes lexicaux en fonction du genre du texte. Nous illustrons cette situation à l'aide d'un exemple curieux, extrait des résultats du fonctionnement du système de TA russe-anglais, dont l'objet était la traduction des textes de caractère général et politique (plus concrètement, les nouvelles de l'agence ITAR-TASS)

L'énoncé (3) V *1999 godu v FRG pereexalo 95 tysjač etničeskix nemcev* a été traduit comme (3a) *In 1999 in the Federal Republic of Germany 95 thousand ethnic Germans were run over*. Cette

traduction, qui heureusement ne correspond pas à la réalité, est en principe légitime, puisqu'elle réalise une SSynt bien formée (dont le sommet est le verbe impersonnel PEREEXAT'2 'écraser', prenant le syntagme *95 tysjač etničeskix nemcev* comme complément d'objet direct). Ce résultat est pourtant facile à éviter : il suffit de baisser la priorité de ce verbe familier par rapport à PEREEXAT'1 'changer le lieu de résidence'.

3.3.2. Renforcement des homonymes lexicaux faisant partie des collocations

Dans les dernières sous-étapes de l'AnalyseSynt, on utilise souvent les règles qui augmentent la priorité des homonymes lexicaux s'il est probable que ces derniers participent dans des collocations. Par exemple, si l'énoncé contient la collocation *nanosit' poraženie* lit. 'faire subir une défaite à quelqu'un', il est raisonnable d'augmenter la priorité du verbe *nanosit'*2 (V-support) et celle du nom *poraženie*2 'défaite' par rapport aux autres acceptions de ces vocables.[2] Dans de telles situations, aussitôt la sous-étape 2 de l'AnalyseSynt terminée, on applique la règle d'augmentation de priorité des homonymes lexicalement liés, qui exploite de façon active la notion de fonction lexicale de Igor Mel'čuk (cf., par exemple, Mel'čuk 1974) et qui s'appuie sur le fait qu'un lien hypothétique nécessaire ait été établi entre les homonymes correspondants. La même technique est utilisée également dans les cas où il est nécessaire d'augmenter la priorité d'une lexie entrant dans un syntagme terminologique. Par exemple, l'adjectif abstrait *ispolnitel'nyj*1 'exécutif' se voit assigner une priorité plus élevée que l'adjectif qualificatif *ispolnitel'nyj*2 'efficace' lorsqu'il fait partie des termes comme *ispolnitel'nyj direktor* 'directeur exécutif', *ispolnitel'nyj sekretar* 'secrétaire exécutif', etc.

3.3.3. Renforcement/affaiblissement des hypothèses syntaxiques

Donnons un exemple de règle qui affaiblit certaines hypothèses syntaxiques faites par l'algorithme de l'AnalyseSynt. Nous allons encore une fois faire appel au système de traduction russe-anglais de l'ETAP-3. Lors du traitement de l'énoncé (4) *V Peterburge otmetili godovščinu vosstania dekabristov 1825 goda*, le système a proposé, comme une première variante, une traduction assez énigmatique : (4a) *Approximately 1825 Decembrists of the year have celebrated in Petersburg an anniversary of uprising*.

Cette traduction correspond à une SSynt parfaitement légitime, où les mots-formes *dekabristov* et *1825* sont liés par la relation syntaxique quantitative-approximative, caractéristique des constructions familières de type *litrov pjat'* 'environ 5 litres'.

Nous n'allons quand-même pas pécher contre la vérité si nous posons que cette construction ne doit pas contenir un numéral écrit en chiffres. En introduisant la condition correspondante dans la règle qui établit cette relation, on est en mesure d'activer une règle spécifique d'affaiblissement des hypothèses syntaxiques, s'appliquant à la fin de l'AnalyseSynt. Comme résultat, la traduction (4a) cède la place à la traduction souhaitée (4b) *An anniversary of uprising of Decembrists of 1925 has been celebrated in Petersburg*. La phrase (4a) peut être obtenue quand-même, mais à une étape éloignée de l'analyse. C'est exactement ce résultat que visent les solutions proposées ici.

3.4. Les limites du posssible

En guise de conclusion, nous aimerions mettre le lecteur en garde contre une évaluation trop optimiste de la présente proposition : bien que le système de priorités donne de bons résultats, ses possibilités sont tout de même assez limitées. La raison en est qu'il n'est pas possible de rendre explicites, par de moyens suffisamment simples et formalisables, tous les faits langagiers ayant trait à l'ambiguïté.

[2] Cf. PORAŽENIE1 'lésion' et NANOSIT'1 'couvrir (avec une couche)'

À titre d'illustration, donnons un exemple d'ambiguïté syntaxique facile à résoudre pour un humain mais qui, de tout évidence, ne peut pas être résolue dans le cadre de l'approche présentée ici.

Dans l'énoncé *Na meste požara byl obnaružen mertvym požiloj storož* lit. '[Un] vieux gardien a été trouvé mort sur [le] lieu d'incendie', on voit le mot-forme *мертвым*$_{[Adj]instr.sg}$ 'mort', qui porte de façon non-ambiguë sur *storož* 'gardien' ; cependant, ce mot-forme est aussi une forme de l'instrumental du singulier du nom (adjecfif nominalisé) *mertvyj*$_{[N]}$ '[un] mort'. Par conséquent, du point de vue formel, rien n'empêche l'analyseur d'interpréter ce mot-forme comme un nom et, de ce fait, comme l'Agent du verbe passif *obnaružit'* 'être trouvé'. Le résultat sera une traduction comme *An elderly guard was found on the site of fire by the dead man* (c-est-à dire que le gardien a été trouvé par un mort.) Il n'est pas possible d'indiquer, dans le dictionnaire ou dans des règles, qu'une expression qui veut dire 'un mort' ne peut que rarement désigner un Agent. Une telle information — qui, de façon générale, caractérise la cooccurrence restreinte — devrait entre autres contenir un renvoi au type sémantique du verbe qui admet un Agent. (Un mort ne peut pas trouver quelqu'un, mais il peut très bien faire peur à quelqu'un ou étonner quelqu'un.)

De cette façon, même dans des cas où le chercheur reste dans le cadre de la tâche modeste qu'il s'est donnée et essaie d' « échelonner » les analyses qui n'ont pas de façon évidente la même probabilité, c'est loin d'être toujours possible. À notre avis, cet état de choses ne remet aucunement en question le système de priorités comme tel, mais permet de cerner de plus près les limites de son applicabilité.

Références

Апресян и др. 1989: Апресян, Ю.Д., И.М.Богуславский, Л.Л.Иомдин, А.В.Лазурский, Н.В.Перцов, В.З.Санников, Л.Л,Цинман. Лингвистическое обеспечение системы ЭТАП-2. М.: Наука, 1989.

[Apresjan *et al*., 1989 : Apresjan, Y. D., I. M. Boguslavskij, L.L. Iomdin, A. V. Lazurskij, N.V. Percov, V.Z. Sannikov, L. L. Cinman. Bases linguistiques du système ETAP-2. Moscou : Nauka, 1989.]

Апресян и др. 1992: Апресян, Ю.Д., И.М.Богуславский, Л.Л.Иомдин, А.В.Лазурский, Л.Г.Митюшин, В.З.Санников, Л.Л.Цинман. Лингвистический процессор для сложных информационных систем. М.: Наука, 1992.

[Apresjan *et al*., 1992 : Apresjan, Y. D., I. M. Boguslavskij, L.L. Iomdin, A. B. Lazurskij, L. G. Mitjušin, V.Z. Sannikov, L. L. Cinman. Un système de traitement de la langue pour les systèmes informatiques complexes. Moscou : Nauka, 1992.]

Богуславский и др. 2000: Богуславский, И.М., Л.Л.Иомдин, Л.Г.Крейдлин, Н.Е.Фрид, И.Л.Сагалова, В.Г.Сизов. Модуль универсального сетевого языка (UNL) в составе системы ЭТАП-3. // Труды международного семинара Диалог'2000 по компьютерной лингвистике и ее приложениям. М., 2000, 48-58.

[Boguslavskij *et al*., 1992 : Boguslavskij I. M., L.L. Iomdin, L. G. Kreidlin, N. E. Frid, I.L. Sagalova, V. G.Sizov. Le module du langage universel des réseaux (UNL) dans le système ETAP-3. // *Actes de colloque international « Dialogue 2000 » sur la linguistique computationnelle et ses applicaitions*, Moscou 2000, 48-58.]

Иомдин-Цинман 1997: Л.Л.Иомдин, Л.Л.Цинман. Лексические функции и машинный перевод. // Труды международного семинара Диалог'97 по компьютерной лингвистике и ее приложениям Диалог'97. М., 1997, 291-297.

[Iomdin-Cinman 1997 : L.L. Iomdin, L. L. Cinman. Fonctions lexicales et traduction automatique. // *Actes de colloque international « Dialogue 1997 » sur la linguistique computationnelle et ses applicaitions*, Moscou 1997, 291-297.]

Мельчук 1974. И.А.Мельчук. Опыт теории лингвистических моделей «Смысл ⇔ Текст». М., Наука.

[Mel'čuk, 1974 :]

Oepen *et al*. 2000: Oepen, Stephan, Dan Flickinger, Hans Uszkoreit, Jun-Ichi Tsujii. Introduction to this Special Issue. // Natural Language Engineering. Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation. 6 (1), 1–14.