SEWS : un serveur d'évaluation orienté Web pour la syntaxe

Olivier Hamon [*,**] — **Patrick Paroubek**[***] — **Djamel Mostefa**[*]

** ELDA, 55-57, rue Brillat-Savarin, 75013 Paris, [hamon, mostefa]@elda.org*
*** LIPN, Université de Paris XIII, 99 avenue J.-B. Clément, 93430 Villetaneuse*
**** LIMSI-CNRS, Bât. 508 Université Paris XI BP 133, 91403 ORSAY Cedex, pap@limsi.fr*

RÉSUMÉ. *Les plates-formes d'évaluation sont à l'heure actuelle peu répandues et les enjeux qu'elles représentent sont certainement sous-estimés, dès lors qu'elles sont automatisées et proposées comme serveur Web : gain de temps, économies de moyens, progression plus rapide des systèmes, rassemblement d'une communauté autour d'un même paradigme d'évaluation; les avantages sont nombreux. Dans cet article, nous proposons une plate-forme pour évaluer automatiquement des systèmes d'annotation syntaxique et relatons son déploiement en tant que service dans une campagne d'évaluation. Après avoir dressé un état de l'art des plates-formes utilisées pour l'évaluation de systèmes TAL et des outils disponibles pour la mise en place de serveurs Web, nous présentons notre plate-forme d'analyse syntaxique et sa mise en œuvre sous forme de serveur Web dans le projet PASSAGE, nous montrons ensuite l'intérêt de la généralisation d'un tel service à d'autres domaines du TAL.*

ABSTRACT. *Examples of Automated Evaluation platforms deployed as Web server are currently very rare and often underestimated. Time and, effort savings, faster system improvement, common paradigm of evaluation for a community, the benefits offered by such services are plentiful. In this paper, we present a platform for evaluating automatically parsers and we comment on its deployment during an evaluation campaign. First, we draw up a state-of-the-art for platforms used in evaluation of NLP systems, then we present the tools available for Web server deployment. Next, we describe our platform and its deployment in the PASSAGE project as a Web server. Finally we show the interest of generalizing such service to other NLP domains.*

MOTS-CLÉS : *évaluation, plate-forme, serveur Web, annotation syntaxique.*

KEYWORDS: *evaluation, platform, server Web, syntactic annotation.*

TAL. Volume 49 – n°2/2008, pages 247 à 270

1. Introduction

Très vite après l'apparition d'Internet, on a vu se développer la notion de serveur de données ouvert à tous par le biais du protocole de transfert de fichier *File Tranfert Protocol* (FTP). Après les serveurs FTP anonymes, ce sont les outils d'indexation de documents comme Gopher qui sont venus remplacer les listes *ad hoc* de sites que les différentes communautés d'internautes s'échangeaient au gré de leurs besoins. Ensuite, l'avènement du premier navigateur Internet, Mosaic, et le développement des moteurs de recherche ont très vite relégué les premiers outils d'indexation dans les couches profondes des moteurs de recherche. Bien entendu, la communauté du traitement automatique des langues (TAL) s'est très vite emparée de ces nouveaux outils, et l'on a ainsi vu apparaître des sites serveurs de corpus (Brown, Suzanne), d'outils (les étiqueteurs Claws, Brill) ou de données linguistiques (les bases de données lexicales WordNet). L'étape suivante a été le développement de premiers services linguistiques permettant de naviguer dans un corpus, une base de données linguistiques ou un dictionnaire (comme par exemple le Trésor de la Langue Française), voire d'utiliser à distance un outil d'analyse comme un étiqueteur morphosyntaxique ou syntaxique, mais ce genre d'utilisation a toujours été plus anecdotique. Après le cataclysme ALPAC, lorsque l'évaluation est réapparue sur la scène TAL, avec les campagnes d'évaluation comparative ouvertes organisées par le NIST et la DARPA sur la transcription automatique de la parole, la compréhension (MUC) ou la recherche documentaire (TREC), les technologies de serveur de données ont été abondamment mises à contribution. Mais jamais à notre connaissance, n'avait été mis en place conjointement dans le cadre d'une campagne d'évaluation l'ensemble des fonctionnalités de service Internet, avant l'exemple que nous présentons dans cet article. SEWS[1] permet en effet de mettre à disposition de tous les participants une plate-forme d'évaluation qui leur permet de s'évaluer à discrétion et d'explorer immédiatement leurs résultats de performance pendant la phase de développement, puis de soumettre leurs données durant la phase de test et d'obtenir rapidement leurs résultats. SEWS concerne l'évaluation des analyseurs syntaxiques, dans le cadre du projet PASSAGE[2] (de la Clergerie, 2007). Les objectifs du projet PASSAGE sont d'améliorer la précision et la robustesse des analyseurs syntaxiques existants pour le français, en les utilisant sur de gros corpus (plusieurs centaines de millions de mots) et d'exploiter les annotations syntaxiques résultantes pour créer des ressources linguistiques plus riches et plus extensives. Les campagnes d'évaluation dans PASSAGE jouent un rôle central dans le sens ou celles-ci permettent l'amélioration intrinsèque des systèmes d'une part et de combiner les différentes annotations syntaxiques des systèmes d'autre part, afin d'obtenir une annotation de référence contenant un minimum d'erreurs.

Outre les problèmes organisationnels, l'évaluation de systèmes par le biais d'une campagne structurée apporte de nombreuses contraintes telles que le temps alloué au développement des systèmes, celui accordé aux participants pour renvoyer les don-

1. Serveur d'évaluation orienté Web pour la syntaxe
2. ANR-06-MDCA-013, http ://atoll.inria.fr/passage

nées de tests, ou encore celui passé à effectuer l'évaluation en elle-même. Les données spécifiques aux expérimentations doivent être délimitées dans un cadre très précis et leur utilisation est sujette à une standardisation liée aux systèmes évalués. Ainsi, une évaluation requiert beaucoup de temps et est coûteuse en ressources. Tandis que les méthodologies d'évaluation peuvent être complexes à mettre en place, le passage à la pratique est, quant à lui, très exigeant. Cela est vrai pour l'évaluation d'un système de traitement automatique de la langue, et ça l'est d'autant plus lorsqu'il s'agit d'organiser une campagne d'évaluation. Le TAL a besoin d'évaluations pour évoluer, mais l'évaluation nécessite la mise en place de protocoles, méthodologies et outils pérennes. La réutilisation d'outils passe par l'emploi de plates-formes qui permettent une continuité dans l'utilisation des ressources employées et aident à la capitalisation des efforts déployés pour une campagne d'évaluation. Il existe de nombreux exemples de plates-formes pour le TAL, l'un des exemples les plus concrets étant la plate-forme LinguaStream (Widlöcher et Bilhaut, 2005) pour l'expérimentation et la visualisation de chaînes de traitement, ou GATE (Bontcheva *et al.*, 2002) qui propose un environnement de développement modulaire pour le TAL. Pourtant, il est étonnant de constater que l'usage de plate-forme s'est peu étendu à l'évaluation des outils de TAL et jamais avant SEWS sous forme de serveur Web. Dans ce qui suit, nous traitons tout d'abord du cadre de l'évaluation dans PASSAGE ainsi que de l'approche employée, puis nous décrivons la manière dont l'infrastructure a été développée, son intérêt et l'apport qu'elle représente. Finalement, nous dressons un bilan par rapport à la campagne d'évaluation qui a été réalisée par le biais de cette plate-forme, du 13 novembre au 21 décembre 2007.

2. Contexte : la campagne d'évaluation PASSAGE

La plate-forme d'évaluation décrite dans cet article s'inscrit dans le cadre du projet PASSAGE (produire des annotations syntaxiques à grande échelle pour aller de l'avant), traitant de l'évaluation des analyseurs syntaxiques du français. Ce projet fait suite aux deux campagnes EASY[3] (Paroubek *et al.*, 2007) du projet EVALDA[4]. Ces premières campagnes ont permis de mettre en place un protocole d'évaluation prenant en compte de manière séquentielle :

– la création d'un large corpus de textes ;

– la définition d'un guide d'annotation ;

– la constitution d'une annotation de référence sur une partie du corpus servant au test des systèmes ;

– l'utilisation du corpus de test par des systèmes participants ;

– l'évaluation des sorties de ces systèmes sur la partie annotée du corpus de test ;

3. Évaluation des analyseurs syntaxiques

4. http ://www.elda.org/rubrique69.html

– l'extension des annotations sur l'ensemble du corpus de test en fusionnant les sorties des analyseurs syntaxiques.

L'évaluation des annotations peut se réaliser indépendamment en constituants (groupe nominal, noyau verbal, etc.) et/ou en relations (sujet-verbe, attribut du sujet, etc.). De manière plus fine, les scores de précision, rappel et f-mesure (Vilnat *et al.*, 2004) sont calculés spécifiquement par annotation (par exemple la relation sujet-verbe) et par genre textuel (journalistique, spécialité, courriel, transcription d'oral, etc.), chaque genre correspondant à un sous-corpus spécifique.

Les mesures d'évaluation admettent 15 relâchements de contraintes différentes, obtenus en combinant les 5 manières de comparer les segments de textes correspondant aux constituants ou cibles de relations, avec les 3 façons de considérer les définitions des constituants (ceux de l'hypothèse, ceux de la référence, ou ceux de l'hypothèse lorsqu'ils existent sinon ceux de la référence).

Fonction	Formule
ÉGALITÉ	$H = R$
FLOU UNITAIRE	$\|H \backslash R\| \leq 1$
INCLUSION	$H \subset R$
INTERSECTION	$R \cap H \neq \emptyset$
BARYCENTRE	$\frac{2*\|R \cap H\|}{\|R\|+\|H\|} > 0.25$

Tableau 1. *Avec H l'empan de texte hypothèse et R l'empan de texte référence, la table donne les formules permettant de comparer les empans correspondant soit aux constituants, soit aux sources/cibles de relations*

La première campagne d'évaluation PASSAGE comprend deux pistes d'évaluation, l'une appelée « EASY classique » et l'autre appelée « PASSAGE ». La première adhère strictement au protocole de la campagne d'évaluation EASY (Paroubek *et al.*, 2005) et fixe pour les corpus une segmentation en mots et en phrases que les participants doivent obligatoirement respecter ; la seconde laisse libre la segmentation en mots et en phrases et repose sur un réalignement avec un algorithme de programmation dynamique (Makhoul *et al.*, 1999) des données fournies par les participants pour calculer par vote majoritaire une segmentation en mots et en phrases commune. De plus, les annotations de référence pour la piste PASSAGE sont elles aussi obtenues en combinant les annotations des participants selon une stratégie de vote majoritaire ; la même qui sera utilisée pour annoter un corpus de plusieurs centaines de millions de mots, résultat de la campagne PASSAGE. Les deux pistes utilisent le format XML avec une DTD[5] dérivée de celle de la campagne EASY.

Le corpus utilisé dans la piste EASY classique est le même que celui de la dernière campagne EASY. Le corpus textuel comprend 40 000 énoncés (généralement

5. Document Type Definition

une phrase) dont un sous-ensemble de 4 000 énoncés annotés a servi de référence lors de la campagne EASY. Le corpus et ces annotations sont connus des participants et utilisés par eux pour améliorer les performances de leur système pendant la phase de développement. À ce corpus de 4 000 énoncés, un complément d'annotations manuelles pour 400 énoncés pris dans la partie du corpus EASY non encore annotée (partie du corpus EASY complémentaire de celle utilisée comme référence dans la campagne EASY) a été ajouté spécifiquement pour la piste EASY classique de la campagne PASSAGE. Ce complément de corpus permet de s'assurer, en partie, que les systèmes n'ont pas été surentraînés sur le corpus de développement en vérifiant à la fin de la phase de développement que les performances obtenues sur l'ancienne référence EASY et celles obtenues sur son complément apporté pour PASSAGE sont corrélées. Bien entendu, une meilleure validation de ce point aurait été obtenue si au lieu de se contenter de compléter les annotations de référence en annotant des énoncés déjà connus des participants nous avions apporté un nouveau matériau annoté au lieu de simplement apporter de nouvelles annotations ; dans une campagne d'évaluation le corpus de test doit toujours être disjoint du corpus de développement. Mais ce critère de validation nous a néanmoins paru suffire car, d'une part, les participants n'ont de leur propre aveu pas eu le temps de modifier leur système depuis la fin de la campagne EASY et, d'autre part, la suite des activités de PASSAGE, après cette première campagne, n'utilisera plus de segmentations en mots et en phrases définies *a priori*. Pour ces raisons, il nous a paru opportun de nous contenter d'un test de validation sur des annotations complémentaires portant sur un matériau connu des participants.

C'est donc la première phase de développement qui est capitale et qui a dès lors nécessité la création d'une plate-forme d'évaluation accessible depuis un serveur Web commun : en effet, un tel processus d'évaluation implique des expérimentations à répétition de la part des participants, devenant très rapidement fastidieuses à effectuer et nécessitant l'installation et l'utilisation de la chaîne d'évaluation.

À tout point négatif son point positif, si nous prenons le risque d'évaluer des analyseurs surentraînés sur le corpus EASY, nous gagnons un temps précieux lors de la phase de test. En effet, le matériau devant être étiqueté par les participants est déjà à la disposition des participants, puis téléchargé par leurs soins sur le serveur commun lors de chaque mesure effectuée pendant la phase de développement ; et ainsi, la phase de test se résume simplement à lancer la chaîne d'évaluation sur le corpus de référence complet réunissant l'ancienne référence EASY et les annotations complémentaires ajoutées pour la première campagne d'évaluation PASSAGE (ce qui peut être fait automatiquement et quasi instantanément).

Cela a un impact non négligeable quant à l'intervention de l'évaluateur, que nous souhaitons minimale dans notre cas (la plate-forme d'évaluation devant être à même de faire un maximum de tâches sans aucune intervention manuelle). Dans un même temps, nous limitons aussi à deux tâches les manipulations humaines des participants nécessaires aux expérimentations sur leurs données de développement :

– envoi des fichiers destinés à être évalués ;

– visualisation des résultats de l'évaluation.

Le risque d'erreur potentiellement introduite par un participant lors du développement en est d'autant réduit.

Pour résumer, le corpus de référence de cette première campagne EASY se décompose en trois parties :

1) le corpus annoté manuellement, utilisé en phase de développement (quantité moyenne, 4 000 énoncés dont le texte et les annotations sont connus des participants) ;

2) le corpus complémentaire annoté manuellement, utilisé en phase de test (quantité faible, 400 énoncés dont seul le texte est connu des participants) ;

3) le corpus non annoté (quantité importante, 40 000 énoncés), qui lui sera utilisé pour réaliser une annotation automatique de l'ensemble du corpus à l'aide de la fusion de l'ensemble des sorties de système selon un algorithme de vote majoritaire.

Dans PASSAGE, 13 systèmes d'analyse syntaxique ont été testés et tous ont apprécié les apports du serveur d'évaluation, à tel point que dès la phase de test terminée, ils ont spontanément demandé à ce que le serveur d'évaluation soit réouvert car ils souhaitaient poursuivre leurs expérimentations avec les corpus de référence et la chaîne d'évaluation.

3. Les interfaces d'évaluation en traitement automatique des langues

Depuis maintenant quelques années, des architectures modulaires et distribuées ont vu le jour comme les environnements GATE (Bontcheva *et al.*, 2002) ou UIMA (Ferruci et Lally, 2004) et peu d'entres elles ont utilisé l'Internet (Cerbah et Daille, 2006). Ce type d'architecture, de service, offre une facilité d'utilisation permettant l'adaptation de différents modules à une même structure.

De cette manière, il a été développé une architecture modulaire (Fanta *et al.*, 2007) dans le cadre du projet européen sur la traduction parole/parole TC-STAR[6], utilisant l'infrastructure UIMA d'IBM[7]. À travers une infrastructure distribuée autour d'un serveur central, trois types de technologies sont successivement prises en compte (reconnaissance de la parole, traduction automatique, synthèse vocale) afin de réaliser une traduction de l'oral vers l'oral. Chaque module est disponible sur un ordinateur distant et est appelé automatiquement. À la fin du processus, les sorties en reconnaissance vocale et traduction automatique sont évaluées automatiquement. Finalement, un rapport d'évaluation est produit, incluant les résultats du processus de traitement et ceux de l'évaluation. Malgré les avantages indéniables d'une telle infrastructure (évaluation automatique, appel automatique des modules, possibilité de conduire de nombreuses expérimentations en utilisant différentes combinaisons de systèmes, en théorie aucun accès aux données d'évaluation pour les participants, etc.), plusieurs critères nous ont

6. http ://www.tc-star.org

7. http ://dl.alphaworks.ibm.com/technologies/uima/UIMA_SDK_Users_Guide_Reference.pdf

dissuadés de retenir cette possibilité : le développement est complexe et long, chaque participant a pour obligation d'adapter son système au formalisme UIMA, l'architecture manque de souplesse et la robustesse demeure limitée par rapport au réseau utilisé.

Pourtant, à notre connaissance et malgré cette expérimentation, il n'existe pas de plate-forme d'évaluation spécifique dédiée à l'évaluation d'outils du TAL telle que nous en donnons la description. En particulier, il semblerait qu'aucune campagne d'évaluation (et encore moins une campagne dont l'objectif soit l'analyse syntaxique) n'ait donné lieu au développement d'une plate-forme permettant des expérimentations multiples évaluées automatiquement, à partir d'un serveur Web commun à l'ensemble des participants.

Pour l'ensemble des sous domaines du TAL, les évaluations se font d'ordinaire manuellement (ou semi-automatiquement), la plupart du temps en appliquant des scripts par ligne de commande. Toutefois, de nombreuses interfaces ont été développées, mais généralement limitées à des jugements humains à appliquer à des sorties de systèmes, ou alors dans un contexte quelque peu différent de celui décrit ici. Ces interfaces permettent alors d'émettre des jugements, voire de calculer des scores, sur les sorties d'un ou plusieurs systèmes, mais elles sont sans réelle interaction avec les participants ou les développeurs de systèmes et ne sont que très peu utilisées *via* Internet. Nous tentons ci-après de fournir des exemples de telles interfaces d'évaluation (bien qu'il ne s'agisse que d'une liste non exhaustive).

En résumé automatique, la plate-forme QARLA (Amigó *et al.*, 2005a) permet l'utilisation de plusieurs métriques automatiques (les mesures QUEEN, KING et JACK), dès lors que la plate-forme reçoit en entrée des références humaines et des sorties de systèmes. Elle a notamment été utilisée lors de la campagne DUC 2004 (Amigó *et al.*, 2005b), puis a été adaptée à la traduction automatique avec la plate-forme IQMT (Giménez et Amigó, 2006) qui inclut quatre différentes métriques du domaine (ROUGE, GTM, METEOR et BLEU/NIST). Toutefois, ces plates-formes nécessitent encore une intervention humaine relativement importante, comme ce serait le cas pour une seule et unique métrique d'évaluation (lancement de scripts en ligne de commande, lecture des fichiers résultats, etc.).

De même que pour IQMT, de nombreuses initiatives ont eu cours en traduction automatique, ce qui se justifie par l'existence de plusieurs métriques automatiques et des possibilités de calculs de scores plus ou moins robustes ne nécessitant pas de jugements humains. EvalTrans[8] (Niessen *et al.*, 2000) est une des plates-formes les plus abouties pour la recherche en traduction automatique. De nombreuses fonctionnalités sont disponibles : possibilité d'ajouter des jugements humains, utilisation de métriques automatiques agissant sur les taux d'erreurs, informations diverses sur les erreurs rencontrées. Du reste, la plate-forme dispose d'une base de données contenant les segments sources, de référence, ainsi que l'ensemble des segments en sortie des systèmes évalués.

8. http ://www-i6.informatik.rwth-aachen.de/web/Software/EvalTrans

Dans le même ordre d'idées, l'utilisation de jugements humains a justifié le développement d'interfaces en traduction automatique, afin de faciliter le calcul des scores et de rendre plus robustes ces jugements. Citons entre autres les interfaces d'évaluation humaine des projets CESTA (Hamon *et al.*, 2006), WMT (Koehn et Callison-Burch, 2007) ou encore IWSLT (Fordyce, 2007).

C'est également en recherche d'information que des interfaces de jugements humains ont été développées de manière similaire, comme certains outils utilisés lors des campagnes CLEF (Nunzio et N.Ferro, 2006). Puisque dans CLEF le même type d'évaluation se réalise sur des corpus de langues différentes, le mieux est alors de réaliser les jugements à travers une même interface, mais à partir de sites différents (un par partenaire de langue différente). L'interface a alors été développée afin d'être utilisée par Internet ; l'ensemble des données, jugements, etc. étant stocké sur un serveur.

En reconnaissance de la parole, une plate-forme pour l'évaluation des systèmes de dialogue, PARADISE (Walker *et al.*, 1997), a permis l'évaluation selon différentes mesures combinées. PARADISE permettait notamment la comparaison de plusieurs systèmes, mais a été peu utilisée, probablement, d'après les auteurs, à cause de sa complexité et de son coût.

Finalement, mis à part pour le projet TC-STAR (et ce de manière limitée), nous n'avons pas trouvé dans la littérature de plate-forme permettant à des participants de déposer leurs données et d'accéder à l'évaluation de leur système sans avoir à effectuer des manipulations complexes. Il s'agit par ailleurs d'un manque majeur en TAL, puisqu'il implique une faiblesse dans la réutilisabilité et le prolongement des activités d'évaluation en communauté, au-delà de la date de clôture d'une campagne. Actuellement, l'évaluation est plutôt vue comme une activité ponctuelle de fréquence relativement faible à cause des ressources requises, et la synergie créée a tendance à retomber une fois la campagne terminée. Or, le serveur que nous avons développé offre un moyen de continuer à faire vivre la communauté constituée lors d'une campagne d'évaluation en pérennisant les activités d'évaluation et en les inscrivant dans un processus de développement des systèmes évalués plutôt que dans un événement à caractère exceptionnel (la phase de test). L'apport du serveur d'évaluation sur la distribution postcampagne de « packages d'évaluation » comme le pratiquent actuellement le NIST ou ELDA, réside dans l'économie apportée sur l'installation du package d'évaluation ; au lieu de devoir l'installer, le gérer et le maintenir sur chaque site participant, son installation n'est alors requise qu'à un seul endroit : sur le serveur d'évaluation. Les participants peuvent donc concentrer leurs efforts sur la tâche essentielle : l'amélioration de leur système tout en bénéficiant d'un retour immédiat concernant les performances de leur système par le biais du serveur d'évaluation.

À notre sens, ce genre d'architecture passe obligatoirement par l'utilisation d'un serveur Web permettant la centralisation des données, des outils de mesure et résultats. De plus, l'utilisation du réseau (en opposition à une utilisation locale) est justifiée par le même type d'arguments qui rendent les services Web indispensables à l'heure

actuelle[9] : c'est le serveur qui conserve les données, qui va vite, et qui dispose des dernières versions des logiciels (les outils de mesure dans notre cas). De plus, le tout étant centralisé, chacun à accès aux mêmes informations au même instant, permettant ainsi de renforcer la synergie entre les participants en autorisant, par exemple si on le souhaite (et de manière potentiellement anonyme), la publication du niveau de performance de chacun, fournissant ainsi à tous une vue d'ensemble leur permettant de se positionner par rapport aux meilleurs résultats.

4. Architecture

4.1. *Principes linguistiques*

La création d'une plate-forme d'évaluation requiert *a priori* de pouvoir « accueillir » les informations provenant de différents systèmes, dans notre cas des analyseurs syntaxiques. Outre les problèmes de nature informatique que concerne la majeure partie de cet article, une telle plate-forme pose le problème de la compatibilité des formalismes utilisés par les différents systèmes. À travers notre cas d'étude qui est l'analyse syntaxique, nous allons voir quels principes sont supposés par la mise en œuvre d'une plate-forme d'évaluation. Faisons d'abord un bref rappel sur le formalisme EASY inspiré de (Carroll *et al.*, 2002) qui a été défini en collaboration avec les participants. Les annotations proposent 6 types de constituants – (1) nominal, (2) adjectival, (3) prépositionnel, (4) adverbial, (5) verbal et (6) prépositionnel-verbal, le dernier étant utilisé pour les verbes à l'infinitif introduits par une préposition – et 14 types de relations fonctionnelles – (1) sujet-verbe, (2) auxilliaire-verbe, (3) c.o.d, (4) complément-verbe, (5) modifieur de nom, (6) modifieur de verbe, (7) modifieur d'adjectif, (8) modifieur d'adverbe, (9) modifieur de préposition, (10) complémenteur, (11) attribut du sujet/objet, (12) coordination, (13) apposition, (14) juxtaposition. Ces annotations sont décrites en détail dans (Vilnat, 2004). Notons qu'EASY ne connaît pas la notion de tête lexicale. Les annotations EASY doivent permettre d'exprimer l'essentiel d'une annotation syntaxique quel que soit son type (de surface ou profonde, complète ou partielle), ceci sans privilégier une approche particulière. Les annotations EASY[10] permettent d'annoter des constituants continus et non récursifs ainsi que des relations représentant les fonctions syntaxiques. Les relations (binaires pour la plupart, la seule relation ternaire étant la coordination) peuvent mentionner comme source ou cible indifféremment des formes individuelles ou des constituants, toutes les opérations de normalisation, transformation et comparaison se faisant par un retour au segment de texte original correspondant à l'objet annoté (constituant ou bien source ou cible de relation). Ce principe fort dans EASY de garder un lien quasi direct avec le texte source dans l'explication des annotations permet de maintenir une relative indépendance dans la gestion des annotations les unes par rapport aux autres. Il est ainsi possible de traiter facilement de manière indépendante une relation particulière,

9. http ://www.generic-nic.net/formation/web-services/web-services.pdf
10. Le guide d'annotation : http ://www.limsi.fr/Recherche/CORVAL/easy

par exemple la relation sujet-verbe, ce qui est très utile pour l'objectif de l'évaluation qui vise à fournir une image détaillée de la manière dont les performances des différents annotations s'organisent, en particulier en fonction du genre de texte traité (robustesse). De même, on peut facilement traiter indépendamment l'évaluation des constituants de l'évaluation des relations, tout en conservant une base commune de comparaison par le biais du retour au texte source, permettant de comparer ainsi un analyseur qui annote uniquement des relations entre mots, avec un analyseur qui n'annote que des constituants, ou avec un analyseur qui lui produit à la fois des constituants et des relations. Un formalisme comme celui de la plate-forme TIGER est moins bien adapté ici, le retour au texte source étant plus compliqué à réaliser, car les annotations peuvent être construites par couches successives, une dépendance étant exprimée par rapport à des dépendances de niveaux inférieurs. En plus d'une relative indépendance « verticale » entre annotations induites par le recours systématique au niveau du texte pour exprimer une annotation, les outils d'évaluation du protocole EASY réalisent une indépendance « horizontale » par le biais de toute une série de relâchements de contraintes sur les frontières des annotations, permettant ainsi d'introduire un certain degré de liberté salutaire pour la prise en compte de phénomènes pour lesquels il n'y a pas d'accord, reconnu, comme la notion de tête par exemple, limitant cette liberté à un niveau compatible avec la réalité linguistique observée dans les corpus (par exemple si on admet la notion de tête lexicale pour les groupes nominaux celle-ci doit se trouver à l'intérieur des frontières du constituant GN correspondant).

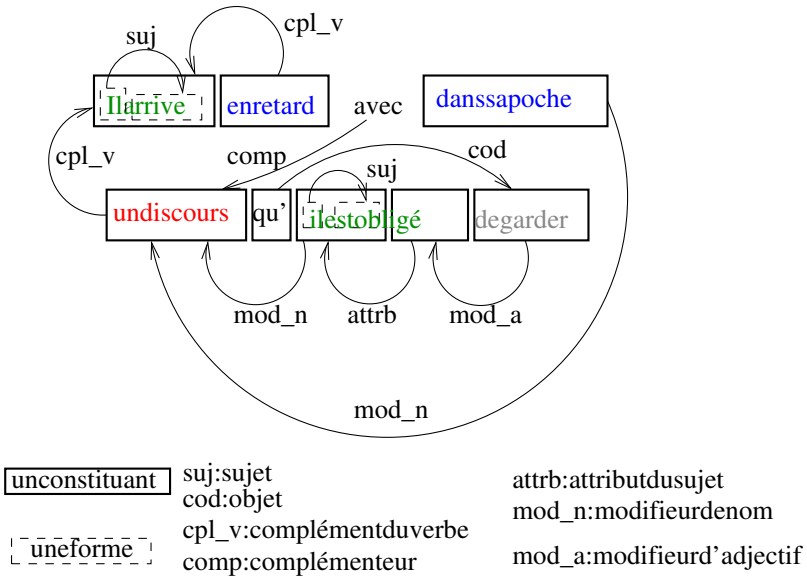


Figure 1. *Exemple d'annotation de l'énoncé « Il arrive en retard, avec, dans sa poche, un discours qu'il est obligé de garder »*

4.2. *Principes informatiques*

La plate-forme d'évaluation décrite dans cet article s'inscrit dans le cadre d'un projet d'évaluation impliquant plusieurs participants. Ce type d'évaluation comporte d'ordinaire beaucoup de contraintes et représente souvent un parcours d'obstacles autant pour les organisateurs que pour les participants. La moindre erreur entraîne bien souvent un regain de temps, requis pour effectuer des corrections, des améliorations, ou tout bonnement pour interagir et communiquer afin de régler les problèmes rencontrés. Ainsi, le protocole doit comporter des actions bien définies, et les différentes validations des données renvoyées par les participants doivent être strictes : les participants devraient avoir un retour d'information pertinent, immédiat et synthétique sur tout dysfonctionnement afin d'éviter les problèmes en cascade et les pertes de temps qui concernent souvent des sujets sans rapport direct à la tâche considérée dans l'évaluation (comme par exemple l'encodage des caractères ou le balisage des données).

D'autre part, et ce toujours afin de limiter le temps passé à réaliser des évaluations, il est important que les participants puissent être autonomes et soient capables d'obtenir eux-mêmes les résultats sur les données de développement.

Pour développer une plate-forme d'évaluation pérenne utilisable par des participants à une évaluation, plusieurs fonctionnalités doivent impérativement être prises en compte :

– l'interface est accessible *via* Internet : pour une évaluation équitable, les participants peuvent accéder aux mêmes données (exception faite de celles inhérentes à leurs systèmes), aux mêmes mesures et aux mêmes types de résultats. Cela permet de conserver en un emplacement unique toutes les données renvoyées par les participants, facilitant ainsi la production d'un historique des variations de performance au cours du développement, un objet très utile pour les tests de non-régression. Enfin, l'accès à la plate-forme d'évaluation *via* Internet permet une utilisation à tout moment, quels que soient la plate-forme et le système d'exploitation utilisés ;

– les données (en entrée et en sortie) sont stockées de façon pérenne sur un serveur d'évaluation ;

– le serveur tient compte de différents types d'évaluations, pour différentes étapes. Comme dans la plupart des campagnes d'évaluations dans le domaine du TAL, PASSAGE comporte deux phases : l'une pour le développement des systèmes, l'autre pour l'évaluation finale. Les actions disponibles pour les participants ne sont pas les mêmes selon le type de phase. De plus, il faut prendre en compte les évaluations ultérieures pour les participants souhaitant poursuivre le développement postcampagne de leur système ;

– la consultation des résultats est intuitive et la plus complète possible. L'exploration des résultats peut s'effectuer avec un grain variable et des outils de visualisation de données peuvent ainsi être partagés entre tous les participants en minimisant les coûts d'installation et de maintenance avec une seule installation sur le serveur d'évaluation au lieu d'une sur chaque site participant ;

– l'ensemble des actions d'évaluation est accompli de la manière la plus automatique possible, avec le moins de manipulation à effectuer pour les participants.

Par ailleurs, le code utilisé pour le développement de la plate-forme d'évaluation doit être réutilisable. En effet, une telle plate-forme nécessite de nombreuses heures de développement et le principe est, à peu de choses près, le même quel que soit le domaine à évaluer (voir le paragraphe 6 sur la généralisation à d'autres domaines).

Plusieurs types d'utilisateurs sont également à prendre en compte :

– les participants : ils effectuent des expérimentations sur les sorties de leurs systèmes au cours de la phase de développement, ils consultent les résultats des évaluations. Ils ont immédiatement accès aux évaluations de la phase de test ;

– les organisateurs : ils peuvent consulter les statistiques d'utilisation du site par les participants et les résultats de la phase de test sur l'ensemble des systèmes participants ;

– les administrateurs du serveur, qui gèrent la base de données et le serveur d'évaluation ;

– les développeurs du serveur : ils ont accès au serveur d'évaluation, et réalisent des tests pour améliorer le site, corriger les éventuelles erreurs, etc. sans pour autant être considérés comme des participants.

Un avantage certain dans le développement d'un serveur d'évaluation, utilisé par plusieurs organismes différents, se trouve dans la nécessité de développer un format commun pour l'interfaçage entre les systèmes et le serveur lui-même. En effet, pour comparer de manière rigoureuse des systèmes, il est nécessaire que l'utilisation des données et mesures soit la même pour tous. Ainsi, les sorties des analyseurs sont transcrites dans un format XML commun, d'après un guide d'annotation clairement défini et validé par l'ensemble des participants. Mais cette utilisation commune d'une même infrastructure dénote l'importance de la confidentialité des données propres à chaque participant. Par-delà cet aspect, c'est la sécurité même d'une plate-forme qui est en jeu lors de son déploiement : les quatre types d'utilisateurs cités plus haut ont des droits bien spécifiques, selon l'utilisation qu'ils auront de la plate-forme. Ainsi, un des points cruciaux d'un serveur Web est la sécurisation des données et des accès. Cela est capital pour plusieurs raisons :

– les participants ne doivent pas *a priori* avoir accès aux données des autres participants (volontairement ou involontairement), mais en fonction d'accords préalables, une visibilité partielle potentiellement anonyme peut être mise en place afin de renforcer la synergie entre les participants ;

– des agents extérieurs ne doivent pas pouvoir corrompre les données ou bloquer l'accès au serveur d'évaluation ;

– les données (en particulier celles des participants) ne doivent pas être perdues ou disséminées.

Cela implique une attention toute particulière lors du développement pour la sécurisation du serveur de données. Ce n'est pas si simple en théorie car il y existe de nombreuses failles de sécurité possibles, plus ou moins faciles à identifier, entre autres : la mauvaise protection des chemin d'accès (URLs, répertoires et fichiers), l'accès à la base de données, le dépassement de capacité mémoire, les injections SQL, les failles CSRF, l'interception de session, etc.

Les droits accordés sont alors progressifs : à la base, aucun utilisateur n'a de droits, puis, au fur et à mesure des autorisations, des droits supplémentaires sont accordés (à l'inverse de droits dégressifs). L'accès est restreint uniquement aux participants s'étant enregistrés au préalable. Ceux-ci n'ont accès en premier lieu qu'aux corpus à annoter. Puis, lorsque les premières soumissions ont lieu, ils ont accès aux résultats de leurs annotations. Pendant ce temps, les organisateurs peuvent avoir connaissance du nombre de soumissions réalisées. Lorsque les tests officiels ont lieu, l'ensemble des évaluations de tous les participants sont visibles de toutes les institutions enregistrées, de manière anonyme pour les participants et de manière identifiée pour les organisateurs.

4.3. *Spécifications et fonctionnalités*

L'objectif du serveur d'évaluation est de permettre à des organismes développant des systèmes d'annotation syntaxique de pouvoir évaluer automatiquement la sortie de leurs systèmes sur les données d'évaluation de la campagne PASSAGE. Pour ce faire, il est nécessaire qu'ils puissent :

– accéder de manière simple au serveur d'évaluation ;

– télécharger facilement le ou les corpus d'évaluation ;

– évaluer automatiquement et de manière indépendante un nombre limité de sorties de système ;

– consulter les évaluations précédemment réalisées ;

– spécifier une « sortie primaire » (par défaut la dernière sortie soumise est retenue) qui servira pour le calcul des résultats de la phase de test (correspondant aux résultats « officiels » de la campagne), jusqu'à la date limite de fin de la phase de développement ;

– faire en sorte que l'accès à la page d'évaluation ne dépasse pas la date limite de la phase de développement ;

– pouvoir, après cette date limite, consulter les évaluations déjà réalisées, sans en effectuer de nouvelles.

Notons qu'à ce stade, un paramétrage permet de définir plusieurs campagnes d'évaluation en parallèle, selon différents critères (phase de développement ou non, limite de soumission, date limite, etc.), permettant ainsi de gérer aisément des « pistes » particulières avec des protocoles et des données modifiées reconfigurables au moyen de paramètres qui permettent la réutilisation de l'infrastructure dans différents cadres d'application.

Les organismes participants ont besoin d'un nom d'utilisateur et d'un mot de passe (spécifique à chaque participant) afin de se connecter au site. Après s'être connectés, ils ont accès à une première page leur permettant de télécharger les corpus d'évaluation, de choisir le type d'évaluation, d'accéder à la page d'évaluation (et la possibilité de soumettre de nouvelles sorties de systèmes) ou encore de consulter leurs évaluations précédentes.

L'utilisation du site est limitée par la durée d'utilisation (fixée à la date limite de la phase de développement) et le nombre d'évaluations possibles (dépendant de la place disponible sur le serveur).

La date limite de soumission pour la campagne PASSAGE était le 21 décembre 2007, à 23 h 59 CET. Dès lors, la soumission d'hypothèses sur le site n'était plus possible après cette date. Au préalable, chaque participant devait avoir identifié sa soumission primaire avant cette date butoir. La soumission primaire était celle utilisée pour comparaison et publication des scores finaux (la dernière soumission est considérée primaire par défaut). Aussitôt la phase de test démarrée, la soumission primaire était évaluée et les résultats envoyés automatiquement aux participants.

L'interface de soumission d'une évaluation est composée de trois parties :

– un champ permettant à l'utilisateur de pouvoir charger sa soumission ;

– une boîte de dialogue permettant à l'utilisateur d'entrer une description de l'expérimentation en cours ;

– un bouton permettant de soumettre le fichier pour être évalué.

L'ensemble des données (site Internet, corpus d'évaluation, corpus de référence, soumissions, outils de calcul des scores) est entreposé sur un serveur de données. L'interface d'évaluation est développée en PHP/MySQL. Les outils de calcul des scores sont contenus dans un module séparé (ici écrit en flex et C++ avec la librairie STL).

Les corpus d'évaluation et de référence sont stockés au préalable sur le serveur. À chaque nouvelle soumission d'un participant, le fichier soumis est également stocké sur le serveur dans un répertoire spécifique. Le format des fichiers soumis est validé automatiquement d'après le format XML, puis un rapport d'erreurs est envoyé au participant, si nécessaire. Ensuite la soumission est évaluée à l'aide de l'outil de calcul des scores. Finalement, les résultats sont également stockés sur le serveur d'évaluation et chaque évaluation réalisée est enregistrée dans la base de données du site. Ces enregistrements permettent de conserver le chemin d'accès aux fichiers soumis, ainsi que les résultats de l'évaluation, les participants pouvant alors consulter les évaluations précédemment réalisées. La page de consultation contient la liste des évaluations, pour chaque évaluation, les fichiers soumis peuvent être téléchargés, ainsi que les résultats de l'évaluation, de même qu'un bouton permettant de sélectionner une soumission comme étant celle qui sera utilisée pour la phase de test (soumission primaire).

Notons que l'affichage des résultats à la suite d'une évaluation peut prendre jusqu'à cinq minutes, et est sujet à évolution en utilisant la technologie AJAX

(Raymond, 2007), qui permet de modifier la partie client – la page affichée dans le navigateur – selon l'évolution côté serveur.

4.4. *Système et structure matérielle*

La partie Web du serveur d'évaluation est implémentée en PHP, tout en utilisant des scripts Javascript pour une meilleure interactivité avec les utilisateurs (tout comme la solution AJAX qui est envisagée pour des versions futures). Mis à part les fichiers XML soumis, les données d'évaluation (informations sur les participants, résultats des évaluations, etc.) sont stockées dans une base de données MySQL. Cette plate-forme est déposée sur un serveur RAID 1 comprenant un processeur Pentium 4 HyperThreading 3,2 GHz, une mémoire de 512 MHz, sous une distribution Linux Debian 4.0. Un espace disque de 120 Go est alloué au serveur d'évaluation et l'interface réseau a un débit descendant de 20 Mo/s et un débit montant de 1 Mo/s.

Mis à part le débit réseau (pour des raisons évidentes d'attente utilisateur), l'espace disque est une des caractéristiques les plus importantes. En effet, la taille des données s'avère rapidement imposante, d'autant que l'ensemble des données (fichiers soumis par les participants, résultats de l'évaluation) est conservé et archivé sur le serveur d'évaluation. Pour une seule soumission, les données en sortie d'un système font environ 120 Mo, les résultats de l'évaluation prenant quant à eux, environ 70 Mo d'espace disque. Dix évaluations de développement étant autorisées en plus de l'évaluation de test, il est donc indispensable de réserver au minimum 30 Go rien que pour la première évaluation PASSAGE, totalisant 13 participants.

De plus, différents outils sont utilisés au sein même de la plate-forme pour les besoins de l'évaluation :

– tar[11] : la commande permet de décompresser automatiquement les archives soumises par les participants, qui peuvent ainsi télécharger sur le serveur des données de taille réduite ;

– xmllint[12] : utilisé pour la validation des fichiers XML soumis par les participants ;

– Gnuplot[13] : utilisé pour l'affichage des résultats en mode graphique ;

– un script de validation des énoncés contenus dans les fichiers XML ;

– l'outil d'évaluation des annotateurs syntaxiques de la campagne EASY ;

– Crontab[14] : planificateur de tâches pour le démarrage automatique de l'évaluation de test.

11. http ://www.gnu.org/software/tar/tar.html
12. http ://xmlsoft.org/xmllint.html
13. http ://www.gnuplot.info/
14. http ://fr.wikipedia.org/wiki/Crontab

Le serveur d'évaluation a été testé au niveau client avec les systèmes d'exploitation Windows (versions 2000, XP), Linux (distributions Debian, Redhat) et Mac Os. Entre autres, les navigateurs Firefox (version 2.0.0.8), Internet Explorer (version 6) et Epiphany (version 2.14.3) ont notamment servi pour ces tests. Le développement du site initial a nécessité environ 80 heures de développement.

4.5. *Infrastructure proposée*

La solution proposée est purement orientée Web : les participants n'ont besoin que d'un navigateur Internet pour pouvoir évaluer les sorties de leurs systèmes, le serveur Web se charge du reste des tâches d'évaluation à effectuer.

On peut décomposer l'architecture de la plate-forme en trois parties, le serveur d'évaluation en étant la partie centrale. En seconde partie, les navigateurs des participants agissent en tant que clients, pour pouvoir télécharger les corpus, entreposer les données annotées, lancer des évaluations et consulter les résultats. La troisième partie concerne le stockage des données, qui sont de trois types : les données des participants collectées, la base de données contenant les informations des évaluations réalisées, et les outils de validation et de mesures de scores (ces derniers étant totalement invisibles aux yeux des participants).

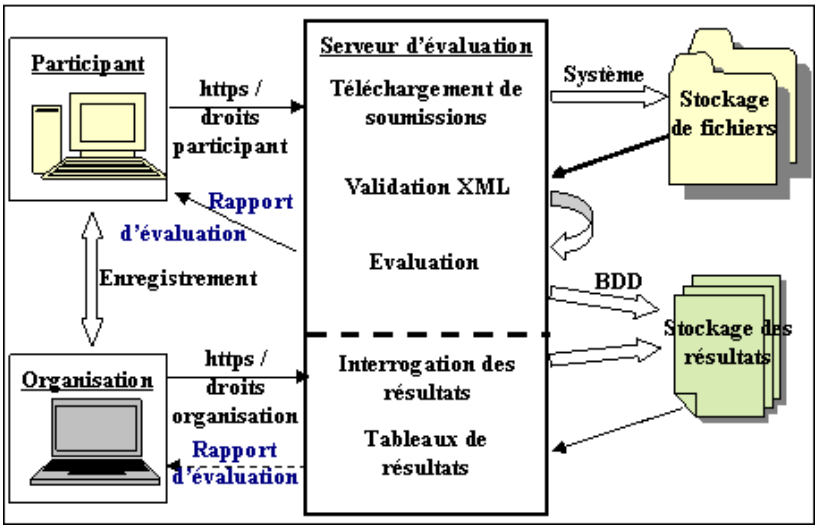


Figure 2. *Architecture de la plate-forme d'évaluation*

La structure principale de la base de donnée est en elle-même relativement simple. Elle est composée de cinq tables :

– table « users » : cette table contient les informations concernant les utilisateurs de la plate-forme (participants et évaluateurs), telles que les identifiants de connexion au serveur d'évaluation, les courriels associés aux participants, le nombre total d'évaluation réalisées, etc. ;

– table « eval_data » : cette table contient les informations sur les données d'évaluation (nom, chemin d'accès et brève description) ;

– table « organisateurs » : cette table contient les informations concernant les organisateurs de la campagne pouvant accéder à la page spécifique récapitulant les données des participants à l'évaluation ;

– table « parser_eval » : cette table stocke les données sur chaque évaluation réalisée par un utilisateur : brève description fournie par l'utilisateur, chemin d'accès, heure et date, récapitulatif des résultats de l'évaluation, etc. ;

– table « primary_eval » : cette table stocke le même type d'informations que la table « parser_eval », mais cette fois-ci pour les évaluations primaires et les résultats obtenus lors de la phase de test.

D'autres tables sont également définies, permettant d'étendre l'utilité du serveur Web d'évaluation à d'autres campagnes d'évaluation (avec ou sans phase de développement). D'une manière générale, il s'agit de tables décrivant les campagnes d'évaluation, les autorisations utilisateur et organisateur pour chaque campagne (tous les utilisateurs ou organisateurs n'ont pas forcément accès à toutes les campagnes d'évaluation, ceci permettant la mise en parallèle de plusieurs projets d'évaluation bien distincts), et les données de campagnes elles-mêmes (dupliquées de la table « parser_eval » décrite ci-dessus).

La collecte des données participant se fait par transfert HTTP, à l'aide de PHP. Un seul fichier est téléchargé sur le serveur, au format compressé targ.gz, tgz ou zip. Ce fichier est placé dans un répertoire correspondant au nom d'utilisateur du participant, ainsi qu'à un identifiant d'évaluation, afin de bien distinguer chaque évaluation.

Après téléchargement de l'archive contenant les fichiers XML d'un participant, elle est décompressé, puis les fichiers XML sont soumis à une validation (le format XML, puis l'ordre des énoncés). Si les fichiers passent correctement la validation, ils sont utilisés dans l'outil de calcul des scores. Pour les besoins d'utilisation, il a fallu modifier l'outil de calcul des résultats afin de pouvoir le combiner correctement avec le serveur d'évaluation, notamment pour des questions d'environnement d'utilisation. Les fichiers résultant de l'évaluation sont placés dans un répertoire similaire à celui des fichiers soumis.

Un dernier point est l'aspect multilingue du serveur d'évaluation, toujours dans l'objectif d'une utilisation plus étendue du serveur Web d'évaluation. Chacune des pages PHP appelle une autre page PHP contenant l'ensemble des données textuelles (titre de la page, textes de menus, etc.). Cette seconde page est située dans un réper-

toire relié à une langue spécifique, dont l'identifiant est contenu dans les variables de session PHP. Cela permet ainsi d'obtenir l'ensemble du site pour une nouvelle langue d'utilisation en ne modifiant que les pages de ce répertoire. Par exemple, lors de l'accès à la page *./evaluation.php*, la page *./langue/english/evaluation.php* sera appelée (si l'utilisateur a choisi l'anglais comme langue de travail) afin d'obtenir le contenu de la page tout en ne modifiant rien à sa structure.

Lorsque l'évaluation est terminée, une page contenant les résultats principaux apparaît (figure 3).

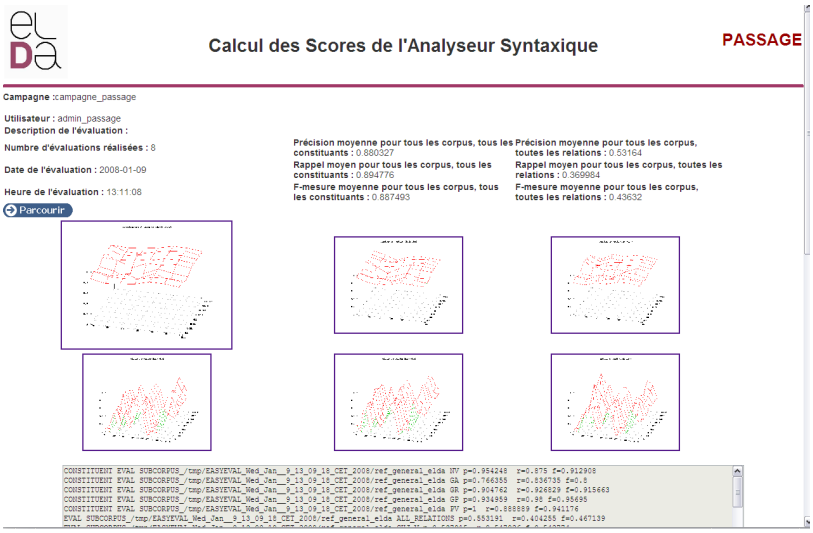


Figure 3. *Présentation des résultats de l'évaluation*

Plusieurs informations sont alors affichées dans un tableau récapitulatif (figure 4) :

– l'identifiant des évaluations stockées sur le serveur ;

– les descriptions des évaluations ;

– la date et l'heure à laquelle les évaluations ont été réalisées ;

– les scores de précision, rappel et f-mesure pour tous les constituants, et sur tous les corpus ;

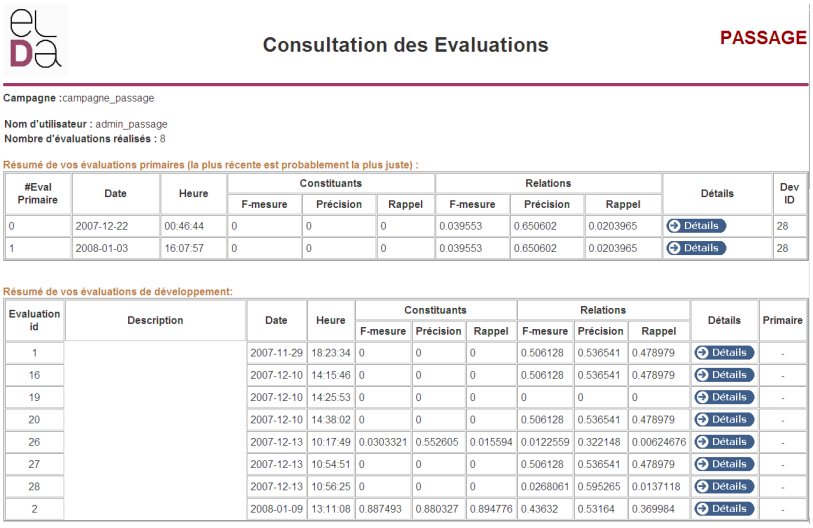
– les scores de précision, rappel et f-mesure pour toutes les relations, et sur tous les corpus ;

– un lien vers les pages fournissant le détail de ces évaluations ;

– une indication sur l'évaluation primaire sélectionnée par le participant ;

– un bouton permettant de supprimer une évaluation.

Le détail de chaque évaluation produit un résultat similaire à la page présentée à la fin de chaque évaluation. Un bouton « Parcourir » y est ajouté, permettant d'accéder et d'explorer les fichiers créés en sortie de l'outil d'évaluation.



Consultation des Evaluations

PASSAGE

Campagne :campagne_passage

Nom d'utilisateur : admin_passage
Nombre d'évaluations réalisés : 8

Résumé de vos évaluations primaires (la plus récente est probablement la plus juste) :

#Eval Primaire	Date	Heure	Constituants			Relations			Détails	Dev ID
			F-mesure	Précision	Rappel	F-mesure	Précision	Rappel		
0	2007-12-22	00:46:44	0	0	0	0.039553	0.650602	0.0203965	Détails	28
1	2008-01-03	16:07:57	0	0	0	0.039553	0.650602	0.0203965	Détails	28

Résumé de vos évaluations de développement:

Evaluation id	Description	Date	Heure	Constituants			Relations			Détails	Primaire
				F-mesure	Précision	Rappel	F-mesure	Précision	Rappel		
1		2007-11-29	18:23:34	0	0	0	0.506128	0.536541	0.478979	Détails	-
16		2007-12-10	14:15:46	0	0	0	0.506128	0.536541	0.478979	Détails	-
19		2007-12-10	14:25:53	0	0	0	0	0	0	Détails	-
20		2007-12-10	14:38:02	0	0	0	0.506128	0.536541	0.478979	Détails	-
26		2007-12-13	10:17:49	0.0303321	0.552605	0.015594	0.0122559	0.322148	0.00624676	Détails	-
27		2007-12-13	10:54:51	0	0	0	0.506128	0.536541	0.478979	Détails	-
28		2007-12-13	10:56:25	0	0	0	0.0268061	0.595265	0.0137118	Détails	-
2		2008-01-09	13:11:08	0.887493	0.880327	0.894776	0.43632	0.53164	0.369984	Détails	-

Figure 4. *Interface de navigation dans différentes évaluations d'un participant*

Nul doute qu'il faudrait également parler ici d'ergonomie. Cela peut avoir son importance dans la perception des résultats, mais surtout cela peut nuire (ou non) à l'impact du serveur d'évaluation sur l'utilité qu'en voient les participants. En effet, l'objectif du serveur est également de faire gagner du temps aux participants (ainsi qu'aux organisateurs) et l'interface peut y jouer un rôle non négligeable. Nous avons choisi d'aller du plus général au plus détaillé, avec deux niveaux de détail, en accédant tout d'abord aux résultats généraux de l'évaluation (scores moyens sur l'ensemble du corpus et des constituants ou relations, graphiques généraux, fichier journal), puis en accédant à l'ensemble des fichiers obtenus en sortie de la métrique d'évaluation et contenant l'ensemble des résultats détaillés. Un troisième niveau de détail existe pour les organisateurs, puisqu'ils peuvent accéder à l'ensemble des résultats de tous les participants à travers une grille d'analyse.

5. Autres méthodes possibles

L'implémentation de cette plate-forme que nous avons choisie n'est pas la seule possible et certains choix ont été faits par facilité, souvent dus aux délais qui nous étaient alors imposés. Malgré cela, les choix effectués sont rationnels et semblent convenir au type de plate-forme développée.

Pour le développement d'une architecture d'évaluation, trois alternatives nous semblent possibles :

– un serveur unique, contenant l'ensemble des données et effectuant les opérations, et des participants passifs (c'est-à-dire n'ayant qu'à envoyer leurs soumissions pour recevoir les résultats d'évaluation) ;

– un serveur ne gérant que la partie évaluation, lié à des applets entreposés chez les participants ;

– un serveur centralisant les informations et répartissant les tâches sur plusieurs autres serveurs dédiés à des tâches bien spécifiques : chaque participant (actif) adapte son système au format du serveur, de même que pour les organisateurs en ce qui concerne la partie métrique d'évaluation ; aucune opération n'est réellement effectuée sur le serveur central.

Nous avons choisi la première solution, les deux autres nous semblant moins adaptées au déploiement d'une campagne d'évaluation pour laquelle les participants auraient à faire le moins d'efforts possibles. En effet, la deuxième solution et surtout la troisième nécessitent plus de temps d'installation. Notons que cette dernière solution est l'approche qui a été retenue dans les infrastructures de type UIMA.

En ce qui concerne le développement en lui-même, plusieurs caractéristiques sont à prendre en compte :

– la gestion des soumissions : nous avons choisi une approche de gestion par fichier plutôt que par base de données, ceci nous semblant plus proche de la tâche – l'effort plus conséquent d'adapter les soumissions à une base de données est très important compte tenu de son utilité relative ;

– l'annotation des fichiers : le serveur d'évaluation utilise la métrique d'évaluation sans se soucier de ce qu'elle prend en entrée, et la validation se fait automatiquement ; l'annotation XML est donc indépendante de la gestion du serveur ;

– le langage de programmation : là également, c'est par facilité que s'est fait notre choix. Tout type de langage est utilisable pour une telle plate-forme en réseau distant, comme par exemple les plus usités : PHP, Python, Perl/CGI, Java, Javascript, C, C++. Le PHP semble être une solution de déploiement plus rapide, mais d'autres langages, comme le C++ ou Java sont certainement plus robustes et permettent, à terme, de meilleurs investissements. Par exemple, l'infrastructure UIMA permet d'associer des modules C++ ou Java par le biais d'une interface XML.

– le protocole de transfert de fichiers : il s'agit de transférer les données les plus importantes ou bien par HTTP (plus exactement HTTPS pour les transferts sécurisés) ou par FTP. Toutefois, la vitesse étant la même que l'on prenne l'un ou l'autre des protocoles, le HTTP semble être le plus adapté, puisque le protocole permet une interaction plus aisée entre le client (côté participant) et le serveur (côté évaluateur), du moins en utilisant le langage PHP.

6. Généralisation

Outre le fait que l'on dispose ici d'un code réutilisable pour étendre la méthode à d'autres domaines, c'est bien le type d'évaluation ouverte et automatique qui apparaît comme le paradigme à appliquer pour d'autres évaluations. Il est intéressant de noter que la quasi-majorité des fonctionnalités peuvent s'appliquer au TAL dans son ensemble, selon un protocole relativement aisé à suivre :

– définition de métriques automatiques d'évaluation (pouvant s'étendre à des métriques fournissant des jugements humains) visant à reproduire l'évaluation des systèmes de TAL ;

– définition d'un corpus d'évaluation assez large pour y contenir des données de développement, de test et de masquage[15] ;

– définition de références (développement, test) applicables aux métriques automatiques précédemment définies ;

– mise en place d'un serveur Web d'évaluation à l'aide des données définies ci-dessus ;

– déroulement des phases d'évaluation (développement, test) selon des dates clairement définies (début de la phase de développement – fin de la phase de développement – début de la phase de test – fin de la phase de test) ;

– multiplication des campagnes d'évaluation sur un même serveur *via* une description dynamique.

A contrario, deux points sont très fortement dépendants du domaine d'évaluation. Tout d'abord, la taille des données d'évaluation qui influe sur le matériel à utiliser. D'une part, certains domaines (comme par exemple la recherche d'information ou la reconnaissance vocale) utilisent de très larges corpus en entrée, ce qui implique la nécessité d'avoir un espace de stockage suffisant. Mais il y a plus important, les systèmes de certains domaines (par exemple en terminologie ou en synthèse vocale) renvoient des sorties elles aussi de dimensions très larges : il faut alors que le débit de téléchargement sur le serveur soit conséquent, et que la taille réservée à l'entrepôt des données le soit également.

L'autre dépendance se trouve dans l'obligation de faire réaliser les évaluations par des experts ou juges humains et nécessite une lourde adaptation du serveur d'évaluation. C'est à notre sens le défaut majeur de la plate-forme présentée dans cet article et l'implémentation future serait très complexe à réaliser. Même si, d'un point de vue technique, la mise en place serait relativement aisée (par exemple en ayant une liste de juges, auxquels des courriels seraient envoyés automatiquement), c'est l'organisation de telles évaluations qui serait particulièrement difficile. En effet, le nombre d'évaluations de développement n'étant pas limité, procéder à des évaluations humaines

15. Les données de masquage sont des données fournies aux participants pour être traitées conjointement avec les données de test, afin que les participants ne puissent identifier sur quelle partie des données ils sont évalués.

reviendrait à des coûts importants et rédhibitoires. La méthode est déjà plus simple à concevoir en termes d'évaluations de test, puisque le jugement d'experts humains ne serait alors requis que pendant cette phase du processus d'évaluation.

7. Conclusion et travaux futurs

La plate-forme d'évaluation a été très utilisée pendant le mois et demi de disponibilité. Au total, 167 évaluations de développement ont été effectuées, sans compter les soumissions erronées et non validées. Les résultats de la phase de test ayant été calculés automatiquement, ils étaient tous disponibles moins d'une heure après la clôture du serveur d'évaluation. Les participants et organisateurs ont pu alors consulter immédiatement leurs résultats. Cela a également permis aux participants de pouvoir disposer en temps réel de toutes les données concernant l'évaluation (données en entrée, données en sortie, fichiers résultats complets), et ce de manière transparente.

Par ailleurs, un certain nombre de requêtes ont été formulées par les participants et organisateurs, visant à améliorer l'emploi du serveur d'évaluation, ce qui ne manquera pas d'être fait pour la seconde campagne du projet PASSAGE. Un des points importants étant l'utilisation du serveur en parallèle (c'est-à-dire de réaliser des évaluations sur les données de développement après la clôture du serveur), l'architecture devant être légèrement remaniée en vue de conserver les résultats des évaluations antérieures tout en réalisant de nouvelles évaluations indépendantes.

De même, la constitution d'annotations de référence étant un problème récurrent (Benzitoun et Véronis, 2005), il est prévu d'améliorer la qualité des références de la phase de test, après cette phase. Il sera donc nécessaire de recalculer les scores des systèmes, ce qui sera facilité par l'existence du serveur d'évaluation.

Dans notre cas, une fois l'infrastructure du serveur d'évaluation créée, il ne reste plus qu'à installer les données requises (données de développement, données de test, informations sur les participants, etc.) et de paramétrer le serveur d'évaluation en fonction du protocole d'évaluation et de la tâche considérée.

Il est nécessaire et largement faisable d'étendre cette approche à d'autres technologies. Elle est simple à mettre en place, quoique coûteuse de prime abord, mais permet la réutilisation ultérieure, outre des données, des scripts et outils d'évaluation sans avoir à refaire des développements, ou reprendre en main des lignes de commandes oubliées de longue date.

La problématique est claire pour une technologie telle que l'évaluation des analyseurs syntaxiques car l'évaluation est automatique. Mais pour d'autres évaluations (recherche d'information, traduction automatique), la mise en œuvre est moins évidente, notamment de par la présence d'évaluations nécessitant des experts ou juges humains. La difficulté de développement s'en ressent, même s'il est possible d'imaginer les différents moyens de parvenir à établir une évaluation cohérente. Entre autres, l'envoi automatique de courriels aux évaluateurs, la mise en place d'un serveur d'éva-

luation à partir de métriques humaines comme cela a déjà été le cas pour d'autres campagnes d'évaluation (en particulier les campagnes CESTA ou TC-STAR pour lesquelles des interfaces d'évaluation s'approchant de cette méthodologie ont déjà été développées).

C'est, à notre connaissance, la première fois qu'une telle expérience de plate-forme d'évaluation a lieu. Cela réflète une évolution des mœurs, mais surtout augure potentiellement un changement de pratique sous des aspects divers : évaluations non ponctuelles ni figées dans le temps, plus de scripts fastidieux à lancer et/ou étudier, médiation de l'évaluation par Internet, etc. C'est donc le mode de fonctionnement qui est en jeu, correspondant par ailleurs aux tests de non-régression qui peuvent se faire en développement de logiciels. L'avenir nous dira, par le biais des utilisateurs de ce genre de plates-formes, si ce mode de fonctionnement est utile ou non. S'il est largement perfectible, nous sommes néanmoins convaincus de son importance future pour de prochaines évaluations de systèmes de TAL.

Remerciements

Nous tenons à remercier les participants au projet PASSAGE pour leurs remarques, conseils et identifications d'erreurs au cours de cette première campagne d'évaluation.

8. Bibliographie

Amigó E., Gonzalo J., nas A. P., Verdejo F., « Evaluating DUC 2004 with QARLA Framework », *Proceedings of the ACL'05 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, Michigan, June, 2005a.

Amigó E., Gonzalo J., nas A. P., Verdejo F., « QARLA : a Framework for the Evaluation of Text Summarization Systems », *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan, June, 2005b.

Benzitoun C., Véronis J., « Problèmes d'annotation d'un corpus oral dans le cadre de la campagne EASY », *Actes des Ateliers de la 12e conférence annuelle sur le traitement automatique des langues naturelles (TALN 2005)*, Dourdan, France, juin, 2005.

Bontcheva K., Cunnihgham H., Tablan V., Maynard D., Hamza O., « Using GATE as an Environment for Teaching NLP », *In proceedings of the ACL Workshop on Effective Tools and Methodologies in Teaching NLP*, Philadelphia, Pennsylvania, p. 54-62, 2002.

Carroll J., Lin D., Prescher D., Uskoreit H., « Toward improved evaluation measures for parsing systems », *Actes de l'atlier « Beyond Parseval » joint à la conférence LREC 2002, 2002.*, Las Palmas, June, 2002.

Cerbah F., Daille B., « Une architecture de services pour mieux spécialiser les processus d'acquisition terminologique », *Traitement Automatique des Langues (TAL)*, vol. 47, n° 3, p. 39-61, 2006.

de la Clergerie E. V., « PASSAGE, Produire des Annotations Syntaxiques à Grande Échelle », *Actes du Grand Colloque STIC 2007*, Paris, France, novembre, 2007.

Fanta M., Fleury P., Kleindienst J., Macek T., « Overview of WP5 : Architecture and Showcases », *Proceedings of the TC-STAR Workshop*, Aachen, Germany, March, 2007.

Ferruci D., Lally A., « UIMA : an architectural approach to unstructured information processing in the corporate research environment », *Natural Language Engineering*, vol. 10, n° 3-4, p. 327-248, 2004.

Fordyce C., « Overview of the IWSLT 2007 evaluation campaign », *Proceedings of the IWSLT 2007 : International Workshop on Spoken Language Translation*, October, 2007.

Giménez J., Amigó E., « IQMT : A Framework for Automatic Machine Translation Evaluation », *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy, May, 2006.

Hamon O., Popescu-Belis A., Choukri K., Dabbadie M., Hartley A., Hadi W. M. E., Rajman M., Timimi I., « CESTA : First Conclusions of the Technolangue MT Evaluation Campaign », *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, May, 2006.

Koehn P., Callison-Burch C., « Evaluating Evaluation - lessons from the WMT 2007 shared task », *Proceedings of the MT Summit XI Workshop : Automatic procedures in MT evaluation*, Copenhagen, Denmark, September, 2007.

Makhoul J., Kubala F., Schwartz R., Weischedel R., « Performance measures for information extraction », *Proceedings of DARPA Broadcast News Workshop*, Herndnon VA, February, 1999.

Niessen S., Och F., Leusch G., Ney H., « An Evaluation Tool for Machine Translation : Fast Evaluation for MT Reseach », *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athen, Greece, June, 2000.

Nunzio G. D., N.Ferro, « Scientific evaluation of a dlms : A service for evaluating information access components », *Proceedings of the 10th European Conference for Digital Libraries (ECDL 2006)*, Alicante, Spain, September, 2006.

Paroubek P., Robba I., Vilnat A., Pouillot L.-G., « Easy : Campagne d'évaluation des analyseurs syntaxiques », *Actes des Ateliers de la 12e Conférence annuelle sur le traitement automatique des langues naturelles (TALN 2005)*, Dourdan, juin, 2005.

Paroubek P., Vilnat A., Robba I., Ayache C., « Les résultats de la campagne EASY d'évaluation des analyseurs syntaxiques du français », *Actes de la 14e Conférence annuelle sur le traitement automatique des langues naturelles (TALN 2007)*, Toulouse, juin, 2007.

Raymond S., *Ajax on Rails*, O'Reilly, janvier, 2007. ISBN 10 :0-596-52744-6.

Vilnat A., Paroubek P., Monceaux L., Robba I., Gendner V., Illouze G., Jardino M., « The ongoing Evaluation campaign of Syntactic Parsing of French : EASY », *Proceedings of LREC*, Lisboa, Portugal, p. 2023-2026, June, 2004.

Walker M., Litman D., Kamm C., Abella A., « PARADISE : A Framework for Evaluating Spoken Dialogue Agents », *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL 1997)*, Madrid, Spain, July, 1997.

Widlöcher A., Bilhaut F., « La plate-forme LinguaStream : un outil d'exploration linguistique sur corpus », *Actes de la 12e Conférence annuelle sur le traitement automatique des langues naturelles (TALN 2005)*, Dourdan, Juin, 2005.