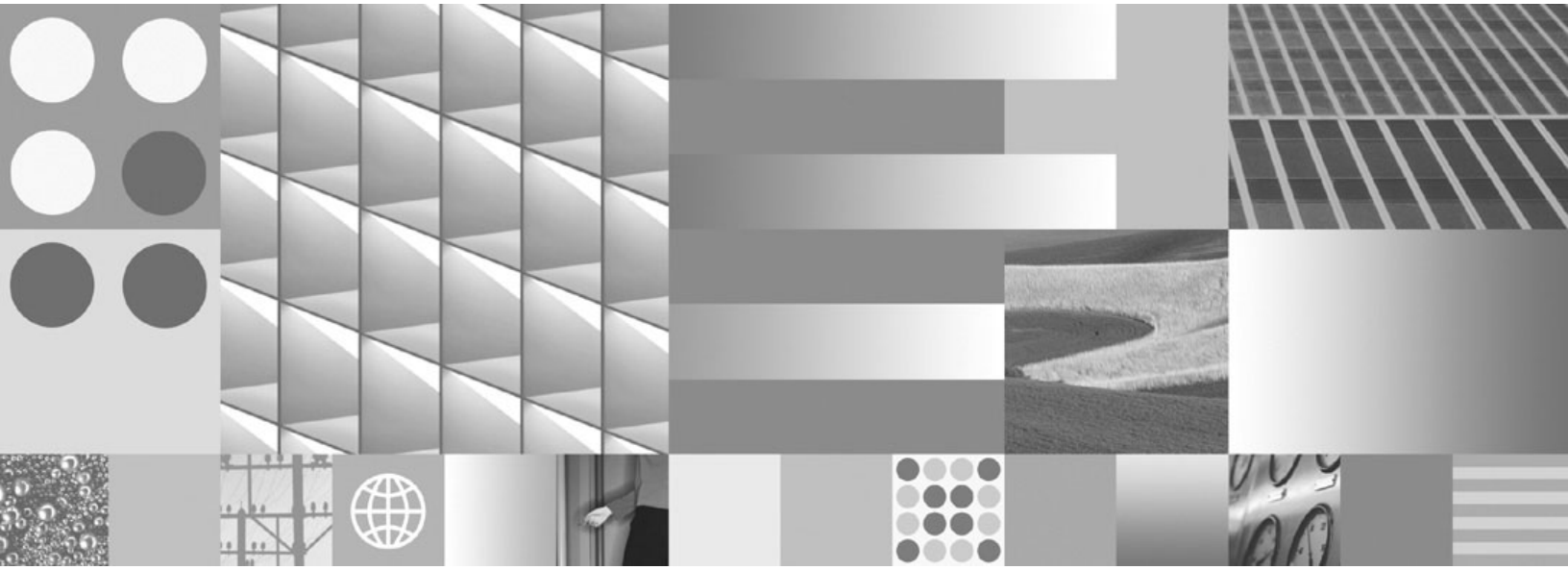
IBM OmniFind Enterprise Edition

Version 8.5

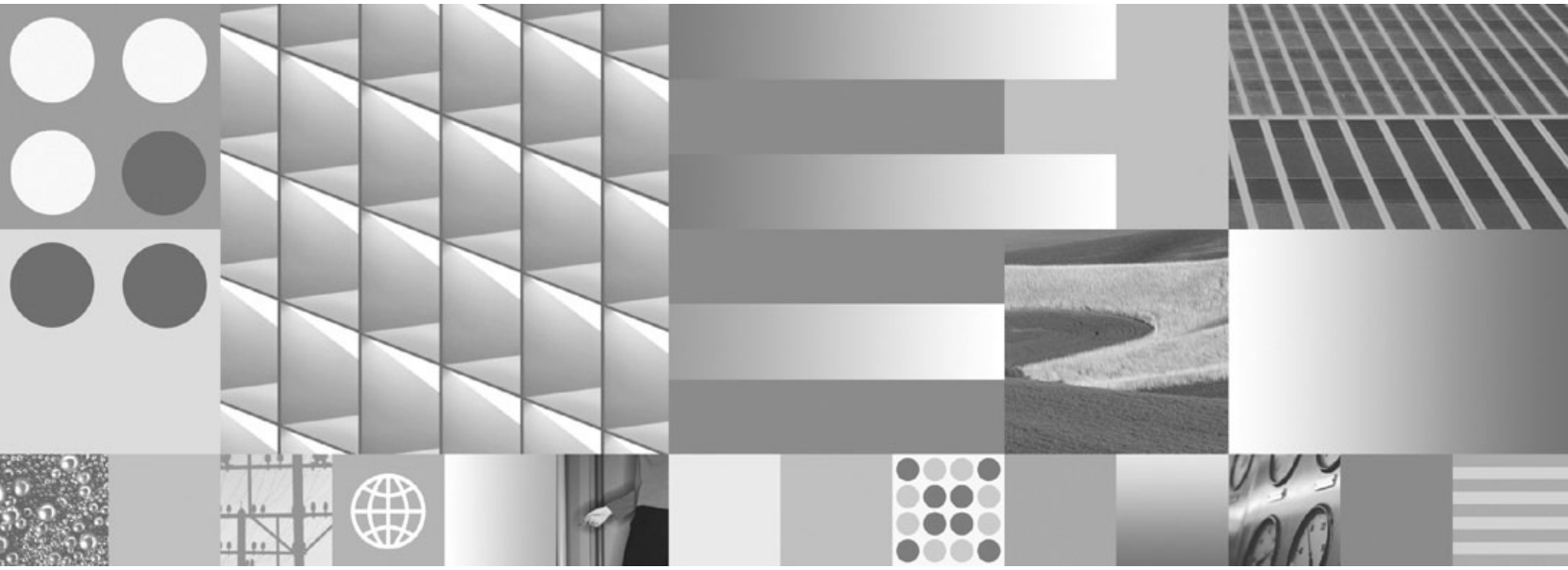


Intégration de l'analyse de texte

SC11-2398-02

IBM OmniFind Enterprise Edition

Version 8.5



Intégration de l'analyse de texte

SC11-2398-02

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant dans la section «Remarques et marques», à la page 117.

Troisième édition - février 2008

Réf. US : SC18-9674-02

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- http://www.fr.ibm.com (serveur IBM en France)
- http://www.can.ibm.com (serveur IBM au Canada)
- http://www.ibm.com (serveur IBM aux Etats-Unis)

Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50

Table des matières

ibm.com et les ressources liées

Vous trouverez la documentation et les informations de support relatives aux produits IBM sur le site ibm.com.

Support et assistance

Les informations de support sont accessibles sur Internet.

IBM OmniFind Enterprise Edition
http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/support.html

IBM OmniFind Discovery Edition
http://www.ibm.com/software/data/enterprise-search/omnifind-discovery/support.html

IBM OmniFind Yahoo! Edition
http://www.ibm.com/software/data/enterprise-search/omnifind-yahoo/support.html

Centre de documentation

Vous pouvez afficher la documentation relative au produit dans un centre de documentation basé sur la technologie Eclipse, via un navigateur Web. Pour consulter ce centre de documentation, accédez à l'adresse suivante : http://publib.boulder.ibm.com/infocenter/discover/v8r5m0/.

Publications PDF

Vous pouvez consulter les fichiers PDF en ligne, via le logiciel Adobe Acrobat Reader. S'il n'est pas installé, téléchargez-le depuis le site d'Adobe, à l'adresse suivante : http://www.adobe.com/fr.

Pour obtenir des publications au format PDF, consultez également les sites suivants :

Produit	Adresse du site Web
OmniFind Enterprise Edition, version 8.5	http://www.ibm.com/support/docview.wss?rs=63&uid=swg27010938
OmniFind Discovery Edition version 8.4	http://www.ibm.com/support/docview.wss?rs=3035&uid=swg27008552
OmniFind Yahoo! Edition version 8.4	http://www.ibm.com/support/docview.wss?rs=3193&uid=swg27008932

Procédure d'envoi des commentaires

Vos commentaires nous permettent de vous proposer des informations toujours plus précises et plus utiles.

Envoyez-nous vos commentaires via le formulaire prévu à cet effet, que vous trouverez à l'adresse suivante : https://www14.software.ibm.com/webapp/iwm/web/signup.do?lang=fr_FR&source=swg-rcf.

Pour contacter IBM

Pour contacter le service client IBM en France, composez le 0810 TEL IBM (0810 835 426).

Pour connaître les options de service disponibles, contactez IBM aux numéros suivants :

- Aux Etats-Unis : 1-888-426-4343
- Au Canada : 1-800-465-9600

Pour plus d'informations sur la procédure à suivre pour contacter IBM, accédez à la page Contactez IBM, à l'adresse suivante : http://www.ibm.com/contact/fr/.

Support linguistique pour la recherche sémantique

La recherche d'entreprise offre un support de recherche linguistique pour les documents texte dans la plupart des langues indo-européennes et dans les langues asiatiques, y compris le japonais.

Vous pouvez utiliser la prise en charge linguistique pour améliorer la qualité des résultats de recherche.

Le traitement linguistique s'effectue en deux étapes : lors de l'ajout d'un document à l'index ou lorsqu'un utilisateur entre une requête de recherche.

La recherche d'entreprise inclut uniquement la fonctionnalité linguistique granulaire ou de base, utilisée pour déterminer la langue d'un document d'entrée et pour segmenter le flux d'entrée du document en mots ou en marqueurs sémantiques.

Si vous savez que les recherches seront restreintes aux recherches de mot clé ou aux recherches XML natives qui utilisent la structure du document, le traitement linguistique inclus dans la recherche d'entreprise est parfaitement adapté à vos besoins.

La plupart des informations des documents texte ne sont pas structurées, ce qui complique leur utilisation efficace car il est difficile d'accéder à leur signification.

La recherche de mots clés est simple, mais elle n'est pas toujours satisfaisante si vous voulez rechercher au-delà des simples mots du document, comme l'illustrent les exemples suivants :

- Dans les cas de collaboration, les informations ne sont pas toujours explicitement marquées, comme une adresse ou un numéro de téléphone dans un courrier électronique. En fait, le terme *numéro de téléphone* peut ne pas être utilisé du tout. Au lieu de cela, le courrier électronique peut contenir une phrase telle que "vous pouvez me joindre au 555-641-1805". Souvent, l'utilisateur ne sait pas sous quelle forme les informations qu'il cherche sont présentées dans le document et voudrait idéalement entrer une requête telle que "numéro de téléphone de Barbara" s'il cherche le numéro de téléphone d'une personne appelée Barbara. Toutefois, cette requête ne réussira pas car le mot *numéro de téléphone* ne se trouve pas dans le document.
- Dans le processus de veille à la concurrence, les documents mentionnent des concurrents et les biens qu'ils fournissent. Ils indiquent également que le site Web d'un concurrent est passé d'une gamme de produits à la vente à une autre au cours des trois derniers mois. Dans ce cas, l'utilisateur peut entrer une requête telle que "biens de Smith & Co." ou "biens de Smith & Co. de nov 2004 à jan 2005". Dans la première requête, le terme *biens* représente un produit ou une gamme de produits, mais la requête ne retournera pas les produits fournis par Smith & Co. car elle recherche le terme *biens*. Il se passe la même chose avec la requête incluant une période particulière. Il est pratiquement impossible d'effectuer une requête sur une période en utilisant une recherche par mot clé.
- Dans la gestion de la relation client, des documents peuvent indiquer des problèmes liés aux freins automobiles dans des ateliers de réparation de la région de San Francisco. Les rapports de l'atelier décrivent des situations telles que "patin réglé à cause d'une fuite hydraulique". L'utilisateur effectuant une

requête pour obtenir plus de détails peut entrer une requête telle que ″ateliers de réparation pour problèmes de freins au nord de San Francisco″. Toutefois, cette requête peut ne pas retourner de rapport relatif au ″patin réglé à cause d'une fuite hydraulique″ car les termes *problèmes de freins* ou *ateliers de réparation* en tant que tels n'apparaissent pas dans les rapports. De plus, ces rapports peuvent mentionner uniquement la ville et le nom du quartier de l'atelier de réparation, pas l'adresse complète avec le nom de la ville San Francisco.

- Dans la recherche, les documents décrivent un médicament particulier, commercialisé à grande échelle sous diverses marques et sa relation à une maladie, au moins, mentionnée dans le même paragraphe. L'utilisateur occasionnel peut entrer une requête à l'aide de l'un des mots connus pour le médicament en espérant obtenir plus de détails sur les diverses maladies, avec leurs symptômes. Toutefois, la requête peut ne pas retourner de documents satisfaisants car le terme connu peut ne pas toujours être utilisé dans les documents et ces documents ne mentionnent généralement pas le mot *maladie*, mais uniquement le nom de la maladie elle-même.

Dans ces exemples, la recherche des éléments requis dans les vastes collections de sources d'informations qui existent aujourd'hui constitue de nouveaux défis pour lesquels une analyse sophistiquée est nécessaire. Cette analyse va au delà du niveau de segmentation et de l'analyse à l'aide de dictionnaires proposés dans la recherche d'entreprise. La plupart des informations intéressantes ne sont pas marquées dans le document d'origine. A la place, le contenu du document doit être analysé afin de reconnaître et trouver des sujets d'intérêt, par exemple, des entités nommées, telles des personnes, des entreprise, des emplacements, des fonctions et des produits ainsi que des relations possibles entre ces entités.

Les informations que vous voulez découvrir et extraire des documents texte sont spécifiques à l'utilisateur et au domaine. Pour vous aider à concevoir et à développer vos propres algorithmes d'analyse, IBM propose l'IBM Unstructured Information Management Architecture (UIMA), architecture et structure logicielle vous permettant de créer des fonctions d'analyse avancées pour trouver des informations d'intérêt dans des collections de documents de la recherche d'entreprise.

Concepts associés

«Intégration de l'analyse de texte personnalisée», à la page 3

«Concepts de base utilisés dans le traitement de l'analyse de texte», à la page 4

Intégration de l'analyse de texte personnalisée

Une fois votre analyse personnalisée générée en dehors de la recherche d'entreprise à l'aide de l'architecture UIMA (Unstructured Information Management Architecture), vous pouvez intégrer la logique d'analyse à la recherche d'entreprise à l'aide de la console d'administration de la recherche d'entreprise.

UIMA est une plateforme ouverte qui identifie les composants pour chaque fonction d'analyse distincte. Elle garantit que ces composants peuvent être facilement réutilisés et associés.

L'analyse linguistique avancée peut inclure une association des différentes tâches d'analyse. L'analyse commence par la détection de la langue et la segmentation et se poursuit par la reconnaissance de la classe des mots suivie d'une analyse grammaticale approfondie. Les dernières tâches incluent l'identification, par exemple, de la relation entre des substances chimiques et des symptômes particuliers. Chaque étape du processus d'analyse dépend des résultats de l'étape précédente.

La logique d'analyse de chaque étape est contenue dans un *annotateur*. Les annotateurs se regroupent pour former une chaîne de traitement itérant sur chaque document de la collection pour découvrir de nouvelles informations et les stocker pour le traitement en aval.

Les annotateurs responsables de la découverte et de la représentation du contenu d'analyse dans les documents texte sont contenus dans un *moteur d'analyse*, un concept central de l'architecture UIMA. Un moteur d'analyse peut contenir un seul annotateur ou peut être composé de plusieurs moteurs, chacun à son tour contenant des annotateurs.

L'architecture UIMA fournit uniquement les blocs de génération de base permettant de créer, de tester et de déployer vos propres moteurs d'analyse. Elle ne fournit pas de fonctionnalité d'analyse linguistique sous la forme de moteurs d'analyse préconfigurés que vous pouvez déployer dans votre environnement UIMA. Toutefois, le traitement linguistique appliqué dans la recherche d'entreprise est disponible sous la forme d'un ensemble d'annotateurs que vous gérez dans UIMA.

Pour pouvoir utiliser UIMA, vous devez installer le SDK UIMA. Le kit de développement est disponible sur IBM developerWorks. Pour obtenir plus d'informations, reportez-vous à la zone WebSphere Information Integrator à l'adresse suivante http://www.ibm.com/developerworks/db2/zones/db2ii/. Le SDK UIMA inclut une implémentation Java de la structure UIMA pour l'implémentation, la description, la composition et le déploiement des composants UIMA.

Le SDK UIMA fournit également un ensemble d'outils et d'utilitaires pour gérer l'architecture UIMA dans un environnement de développement Eclipse (modules d'extension Eclipse). Pour plus d'informations sur Eclipse, reportez-vous à `www.eclipse.org` et à la documentation UIMA pour obtenir des instructions sur l'installation du SDK UIMA dans l'environnement de développement interactif Eclipse.

Concepts associés

«Support linguistique pour la recherche sémantique», à la page 1

«Concepts de base utilisés dans le traitement de l'analyse de texte»

Concepts de base utilisés dans le traitement de l'analyse de texte

Les concepts de base utilisés dans le traitement de l'analyse de texte incluent les annotateurs, les résultats d'analyse, la structure de fonctions, le type, le système de types, l'annotation et la structure d'analyse commune.

Les *annotateurs* contiennent la logique qui analyse un document et découvre et enregistre les données descriptives relatives au document dans son ensemble (appelées métadonnées du document) et les parties du document. Ces données descriptives sont appelées *résultats d'analyse*. Les résultats d'analyse annotent toute sous-chaîne contiguë (également appelée étendue) du document texte. Dans l'idéal, les résultats de l'analyse correspondent aux informations que vous recherchez.

Une *structure de fonctions* est la structure de données sous-jacente qui représente un résultat d'analyse. Une structure de fonctions est une structure attribut-valeur. Chaque structure de fonctions est d'un *type* et chaque type a un ensemble défini de fonctions ou d'attributs valides (propriétés), tout comme une classe Java. Les fonctions ont une plage qui indique le type de valeur que la fonction doit avoir, par exemple String.

Par exemple, l'étendue texte ″James Matthew Bloggs″ peut être étendue par une annotation de type `Person` avec les fonctions `personName`, `age`, `nationality` et `profession`.

Le *système de types* définit les types d'objets (structures de fonctions) pouvant être découverts dans un document. Le système de types définit toutes les structures de fonctions possibles en termes de types et de fonctions (attributs), de la même manière qu'une hiérarchie de classes dans Java. Vous pouvez définir autant de types différents que vous le souhaitez dans un système de types. Un système de types est spécifique à un domaine et à une application.

La plupart des annotateurs d'analyse de texte produisent leurs résultats d'analyse sous forme d'*annotations*. Les annotations sont un type spécial de structure de fonctions conçue pour le traitement de l'analyse linguistique. Une annotation s'étend sur un texte d'entrée et est définie en termes de positions de début et de fin du texte d'entrée.

Par exemple, un annotateur qui reconnaît des expressions monétaires crée pour le texte ″100,55 dollars des Etats-Unis″ une annotation de type `monetaryExpression` qui couvre le texte avec la fonction `currencySymbol` ayant la valeur ″$″.

Tous les annotateurs de l'architecture UIMA modélisent et stockent les données dans des structures de fonctions.

Toutes les structures de fonction sont représentées dans une structure de données centrale appelée *structure d'analyse commune*. Toutes les données d'échange sont gérées par l'utilisation de la structure d'analyse commune.

La structure d'analyse commune contient les objets suivants :

- Le document texte
- Description du système de types qui indique les types, les sous-types et leurs fonctions
- Résultats de l'analyse qui décrivent le document ou des régions du document
- Référentiel d'index qui prend en charge l'accès aux résultats de l'analyse et les itérations de ces derniers

Concepts associés

«Support linguistique pour la recherche sémantique», à la page 1

«Intégration de l'analyse de texte personnalisée», à la page 3

Algorithmes d'analyse de texte

Le SDK UIMA inclut des API et des outils qui permettent de créer des annotateurs (algorithmes d'analyse incluant la description du système de types) et d'intégrer ces annotateurs dans les moteurs d'analyse.

La documentation UIMA inclut un tutoriel qui vous guide dans le processus de génération de ces composants. Le SDK inclut des utilitaires de test et d'affichage des résultats et un moteur de recherche sémantique à petite échelle pour l'indexation des résultats de la recherche. Vous pouvez également effectuer une recherche sémantique plus avancée sur les informations stockées dans l'index.

Parce que le SDK UIMA ne fournit pas d'annotateurs pré-configurés et parce que tout annotateur personnalisé que vous développez à l'aide de l'architecture UIMA et intégrez dans la recherche d'entreprise est généré d'après les résultats des annotateurs de base de la recherche d'entreprise, vous pouvez utiliser le package d'annotateurs de base de votre environnement UIMA. Pour savoir comment inclure la fonction de détection de langue et de marquage sémantique avant d'exécuter vos algorithmes d'analyse de texte personnalisés dans votre environnement UIMA, reportez-vous à la documentation UIMA.

Une fois que vous avez développé et testé vos moteurs d'analyse à l'aide du SDK UIMA, vous devez créer un fichier PEAR (Processing Engine ARchive) pour exécuter vos algorithmes sur une collection de documents de la recherche d'entreprise. Ce fichier d'archive inclut l'ensemble des ressources requises pour le déploiement de la fonctionnalité d'analyse personnalisée en tant que moteurs d'analyse dans la recherche d'entreprise. La procédure de création d'une archive est décrite dans la documentation UIMA fournie dans le SDK.

L'archive que vous créez pour télécharger vers la recherche d'entreprise doit uniquement contenir votre logique d'analyse personnalisée. Elle ne doit pas contenir d'annotateurs de base de la recherche d'entreprise même si votre logique d'analyse personnalisée repose sur les résultats d'annotateurs de base parce que les annotateurs de base s'exécutent toujours avant toute analyse personnalisée dans la recherche d'entreprise.

Pour savoir comment configurer et déployer une solution de recherche sémantique dans la recherche d'entreprise, suivez les instructions du tutoriel mentionné à l'adresse Web http://www.ibm.com/developerworks/db2/zones/db2ii/. Le

tutoriel vous guide dans les procédures permettant de déployer des algorithmes d'analyse de texte personnalisés dans la recherche d'entreprise et vous indique également comment utiliser les résultats de l'analyse dans les requêtes afin d'améliorer les résultats de la recherche.

Tâches associées

«Utilisation des annotateurs de base de la recherche d'entreprise dans l'architecture UIMA», à la page 7

Flux de travaux pour l'intégration de l'analyse personnalisée

Vous créez et testez vos algorithmes d'analyse de texte personnalisés à l'aide du SDK UIMA, puis vous les déployez et les exécutez sur les collections de documents dans la recherche d'entreprise.

Pour développer des algorithmes d'analyse et les intégrer à la recherche d'entreprise, procédez comme suit :

1. Planification et conception :
   a. Déterminez les informations à rechercher. Quels sont les documents que vous souhaitez extraire ? Quels sont les concepts et relations nécessaires pour chaque tâche de recherche particulière ? Par exemple, des noms de produit et d'employé peuvent être nécessaires pour améliorer les recherches d'ordre général sur un site Web interne d'une entreprise pharmaceutique alors que les employés du service de recherche et développement ont besoin d'utiliser des variantes de noms de médicament et voir les relations médicament-cause-soin.
   b. Indiquez le type d'analyse de texte requis pour l'extraction des informations des documents dans lesquels effectuer la recherche.
   c. Si la collection contient des documents XML, déterminez si vous souhaitez exploiter le marquage XML dans votre solution. Dans la recherche d'entreprise, vous pouvez utiliser le marquage XML d'une des manières suivantes :
      - Si vous pouvez utiliser le marquage XML dans votre analyse personnalisée (par exemple, vos documents contiennent des éléments `<summary>` ou `<topic>` pouvant être utiles dans un annotateur de récapitulation ou de catégorisation), créez un fichier de mappage des éléments XML à la structure d'analyse commune.
      - Si vous souhaitez utiliser le marquage XML dans vos requêtes tel qu'il apparaît dans le document, vous devez activer le mappage XML natif.
   d. Déterminez quelles sont les informations du résultat de l'analyse de texte stockées dans la structure d'analyse commune auxquelles vous souhaitez pouvoir accéder à l'aide de la recherche sémantique. Créez un fichier de mappage de la structure d'analyse commune à l'index.
   e. Déterminez si vous souhaitez stocker les résultats de l'analyse dans une base de données relationnelle, par exemple, afin de connaître les tendances et les associations en utilisant des applications d'exploration de données ou de génération de rapports. Créez un fichier de mappage de la structure d'analyse commune à la base de données.
   f. Concevez l'application de recherche sémantique. Déterminez l'utilisation des fonctions supplémentaires de la recherche sémantique. Concevez l'interface utilisateur.
2. Développez les activités du SDK UIMA :
   a. Définissez les étapes de l'analyse individuelle.

b. Décrivez le système de types des mappages et des algorithmes de mappage.
c. Développez les algorithmes d'analyse (annotateurs) pour chaque étape de l'analyse et intégrez les annotateurs aux moteurs d'analyse à l'aide du SDK UIMA. Générez chaque analyse personnalisée à l'aide de la fonctionnalité de base (identification de langue et marquage sémantique) dans le module des annotateurs de base de la recherche d'entreprise.
d. Après avoir testé les algorithmes d'analyse dans UIMA, placez le(s) moteur(s) d'analyse dans un fichier PEAR (Processing Engine Archive). L'archive ne doit contenir que vos algorithmes d'analyse et non la fonctionnalité linguistique de recherche d'entreprise de base.

   Lorsque vous concevez une solution d'analyse de texte, elle peut inclure plusieurs modules d'analyse fournis dans plusieurs fichiers PEAR. L'architecture UIMA permet de fusionner deux ou plusieurs fichiers PEAR en un, que vous pouvez télécharger et exécuter dans la recherche d'entreprise. La fonction de fusion des fichiers PEAR permet d'éviter les conflits de noms, de garantir que les fonctions d'entrée et de sortie sont correctement fusionnées et qu'aucune substitution de paramètre n'a lieu si des paramètres des descripteurs d'annotateurs ont les mêmes noms. Pour obtenir des instructions sur la fusion de fichiers PEAR, consultez la documentation UIMA.

3. Déployez les activités de recherche d'entreprise :
   a. Téléchargez le fichier d'archive (.pear) du moteur de traitement dans la recherche d'entreprise. Attribuez un nom au composant d'analyse de texte afin que vous puissiez y faire référence dans la recherche d'entreprise.
   b. Associez une ou plusieurs collections de documents à votre composant d'analyse de texte.
   c. Si possible, pour chaque collection, téléchargez et sélectionnez le fichier de mappage de l'élément XML à la structure d'analyse commune que vous avez défini pour votre analyse personnalisée.
   d. Si possible, pour chaque collection, téléchargez et sélectionnez le fichier de mappage de la structure d'analyse commune à la base de données que vous avez défini pour votre analyse personnalisée.
   e. Pour chaque collection, téléchargez et sélectionnez le mappage de la structure d'analyse commune à l'index que vous avez défini pour la recherche sémantique.
   f. Lorsque cela s'avère nécessaire, configurez l'application de recherche sémantique personnalisée. Déployez par exemple, l'interface utilisateur de recherche sur navigateur dans un serveur d'applications.
   g. Parcourez, analysez et indexez les documents de la collection de recherche sémantique comme vous le feriez pour une collection ayant recours à des mots clés.

Tâches associées

«Utilisation des annotateurs de base de la recherche d'entreprise dans l'architecture UIMA»

Utilisation des annotateurs de base de la recherche d'entreprise dans l'architecture UIMA

Vous pouvez utiliser les annotateurs du module d'annotateurs de base de la recherche d'entreprise pour développer de nouveaux annotateurs dans le kit de développement de logiciels UIMA et pour mapper les résultats d'analyse aux tables JDBC.

L'ensemble des annotateurs de base inclut :

- **Annotateur d'ID de langue**

  Détecte la langue d'un document. Pour connaître les fonctions et les paramètres de configuration, reportez-vous au fichier du descripteur jlangid.xml.

- **Annotateur de recherche du dictionnaire FROST**

  Permet une création de marqueurs sémantiques et une détection de phrase effectuées en fonction des dictionnaires IBM LanguageWare. Pour les marques sémantiques, des informations linguistiques supplémentaires, par exemple la forme de base ou le lemme, sont générées. Pour connaître les fonctions et les paramètres de configuration, reportez-vous au fichier du descripteur jfrost.xml.

- **Marqueur sémantique d'espace**

  Peut effectuer un marquage sémantique à base d'espaces sur tous les documents de langue européenne ou autres scripts utilisant la séparation par espace. De plus, l'annotateur peut effectuer le marquage sémantique n-gram sur les scripts de texte suivants : arabe, han, hébreu, hiragana, katakana, laotien, mongolien, thaïlandais, yi et hangul. Cette liste contient tous les scripts de texte majeurs asiatiques et signifie que l'annotateur prend en charge le japonais, le chinois ainsi que le coréen.

  Pour connaître les fonctions et les paramètres de configuration, reportez-vous au fichier du descripteur jtok.xml.

- **Annotateur d'expression régulière**

  Détecte les entités ou étendues d'informations dans un document texte basé sur des expressions régulières. Vous pouvez personnaliser l'annotateur d'expressions régulières pour qu'il détecte les entités textuelles dont vous avez besoin en définissant vos propres règles. Un exemple d'annotateur d'expressions régulières qui détecte les numéros de téléphone, les URL et les adresses électroniques dans les documents texte est inclus dans le module d'annotateurs de base.

- **Consommateur de la structure d'analyse commune à la base de données**

  Le consommateur de la structure d'analyse commune à la base de données remplit une base de données relationnelle avec des résultats d'analyse de texte spécifiques.

Le module d'annotateurs de base de la recherche d'entreprise est un fichier compressé (.zip) contenant les annotateurs d'analyse de texte de base, l'annotateur d'expressions régulières et le consommateur de la structure d'analyse commune à la base de données. L'annotateur d'ID langue, l'annotateur de recherche du dictionnaire FROST et le marqueur sémantique d'espaces sont les annotateurs d'analyse de texte de base qui s'exécutent toujours avant toute analyse de texte personnalisée lorsque les documents sont analysés dans la recherche d'entreprise.

Parce que les annotateurs d'analyse de texte de base s'exécutent toujours avant toute analyse de texte personnalisée dans la recherche d'entreprise et parce que toute analyse de texte personnalisée repose sur la sortie des annotateurs de base, vous pouvez utiliser ces annotateurs sur votre environnementUIMA lorsque vous développez et testez vos annotateurs personnalisés.

L'annotateur d'expressions régulières et le consommateur de la structure d'analyse commune à la base de données sont des options supplémentaires que vous pouvez sélectionner sur la console d'administration de la recherche d'entreprise lorsque vous configurez vos options de traitement de texte. Vous pouvez également les utiliser dans UIMA. Pour la personnalisation avancée de l'annotateur d'expressions régulières, nous vous recommandons d'utiliser les outils du kit de développement de logiciels UIMA fourni.

Pour exécuter ces annotateurs dans l'architecture UIMA, le kit de développement de logiciels UIMA doit être installé. Ce dernier est disponible sur le site Web IBM developerWorks à l'adresse http://www.ibm.com/developerworks/db2/zones/db2ii/.

Pour installer le module d'annotateurs dans votre installation du kit de développement de logiciels UIMA :

1. Recherchez le module d'annotateurs OF_base_annotators.zip dans l'installation de recherche d'entreprise (OmniFind Enterprise Edition) se trouvant dans le répertoire *RACINE_INSTALL_RECHERCHE_ENTREPRISE*/packages/uima.
2. Copiez le fichier zip dans le répertoire racine de l'installation du SDKUIMA.
3. Extrayez le fichier zip pour ajouter les fichiers d'annotateurs de base de la recherche d'entreprise à l'arborescence des répertoires indiquée pour votre installation du kit de développement de logiciels UIMA. Le fichier tt_core_typesystem.xml sera remplacé. Si vous souhaitez conserver l'ancienne version de ce fichier, enregistrez-la avant d'extraire le fichier zippé.
4. Pour définir le chemin de classes, ouvrez le script setUIMAClasspath du répertoire bin et ajoutez une ligne à la fin du script qui commence le script OFAnnotEnv.
5. Si vous voulez utiliser des types spécifiques de recherche d'entreprise ou personnalisée dans UIMA, reportez-vous à la documentation du kit de développement de logiciels UIMA pour savoir comment les définir.

Une fois que vous avez installé le module d'annotateurs de base, vous pouvez trouver les fichiers du descripteur d'annotateur dans le répertoire *INSTALL_SDK_UIMA*/docs/examples/descriptors/analysis_engine. Le fichier of_tokenization.xml répertorie les annotateurs d'analyse de texte de base (l'annotateur d'ID langue, l'annotateur de recherche du dictionnaire FROST et le marqueur sémantique d'espaces) dans la séquence dans laquelle ils sont utilisés au sein de la recherche d'entreprise.

Les fichiers de descripteur contiennent les valeurs de configuration autres que celles utilisées dans la recherche d'entreprise. Vous pouvez modifier les valeurs à des fins de débogage dans le kit de développement de logiciels UIMA. Toutefois, ne modifiez pas ces fichiers de descripteur dans votre système de recherche d'entreprise. Le fait d'apporter des modifications à ces fichiers peut générer une instabilité du système ou des incidents de performances.

Le module annotateur de base de recherche d'entreprise contient uniquement les dictionnaires requis pour le traitement des documents anglais. Si vous souhaitez traiter d'autres langues dans votre environnement de développement, suivez la procédure ci-après.

1. Recherchez les dictionnaires de recherche d'entreprise dans l'installation de recherche d'entreprise dans *RACINE_INSTALL_RECHERCHE_ENTREPRISE*/configurations/parserservice/jediidata/frost/resources.
2. Copiez le contenu du répertoire vers votre installation SDK UIMA locale dans *UIMA_SDK_INSTALL*/data/frost/resources.

Pour vérifier que l'installation du module d'annotateur a abouti, procédez comme suit :

1. Ouvrez le débogueur visuel CAS (CVD) dans le répertoire suivant : *INSTALL_SDK_UIMA*/bin/cvd[.bat/.sh].
2. Cliquez sur **Exécuter → charger TAE**.

3. Sélectionnez le fichier d'indicateur d'analyse de texte of_tokenization.xml dans le répertoire *INSTALL_SDK_UIMA*/docs/examples/descriptors/analysis_engine.
4. Chargez un document exemple et exécutez le moteur d'analyse de texte. Vous verrez s'afficher des annotations du type uima.tt.TokenAnnotation dans le CVD.

Si vous exécutez l'un des annotateurs d'analyse de texte de base avant vos annotateurs personnalisés dans votre environnement de développement et que vos annotateurs personnalisés utilisent les types définis par l'analyse de texte de base, incluez une référence au fichier tt_core_typesystem dans la section de système de types de votre spécificateur d'annotateurs personnalisés. Le fichier tt_core_typesystem se trouve dans le répertoire *UIMA_SDK_INSTALL*/docs/examples/descriptors/analysis_engine. Pour obtenir un exemple de la procédure d'inclusion des références dans des fichiers de descripteur, consultez le fichier jtok.xml dans le répertoire analysis_engine.

Tâches associées

«Affichage des résultats de l'annotateur de base et de l'analyse de texte personnalisée», à la page 13

«Activation de la recherche sémantique facilitée à l'aide de l'annotateur d'expressions régulières», à la page 86

«Utilisation du consommateur de la structure d'analyse commune à la base de données dans l'architecture UIMA»

«Utilisation de l'annotateur d'expressions régulières dans l'architecture UIMA», à la page 13

Utilisation du consommateur de la structure d'analyse commune à la base de données dans l'architecture UIMA

Pour pouvoir utiliser le consommateur de la structure d'analyse commune à la base de données dans l'architecture UIMA, vous devez apporter des modifications au fichier du descripteur du consommateur et écrire le fichier de mappage de la structure d'analyse commune à la base de données.

Pour pouvoir exécuter le consommateur de la structure d'analyse commune à la base de données dans votre environnement UIMA, vous devez :

1. Ouvrir le fichier descripteur XML cas2jdbc.xml dans *UIMA_SDK_INSTALL*/docs/examples/descriptors/cas_consumer. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix.
2. Modifier le paramètre **mappingFile** pour inclure le chemin d'accès absolu dans lequel se trouve le fichier de mappage de la structure d'analyse commune à la base de données, par exemple, D:\temp\MyMapping.xml.
3. Modifier le paramètre **docMetadata_Type** pour indiquer le type UIMA à partir duquel toutes les métadonnées des fonctions intégrées sont extraites, par exemple, uima.tcas.DocumentAnnotation.
4. Modifier le paramètre **docId_Feature** pour inclure la fonction ou le chemin de fonctions au type de métadonnées à partir duquel un ID numérique de document (de type nombre entier) est extrait. Ce paramètre est requis par toutes les fonctions intégrées qui requièrent l'ID, telles que, docId(), uniqueId(), objectId() et fsId().
5. Ne pas définir le paramètre **encryptionClass** car il est uniquement utilisé dans la recherche d'entreprise pour permettre au consommateur de la structure d'analyse commune à la base de données de fonctionner avec les fichiers de mappage chiffrés.

6. Enregistrer le fichier.
7. Copier les fichiers de bibliothèque EMF (common.jar, ecore.jar et ecore.xmi.jar) du répertoire `lib` de votre installation de recherche d'entreprise vers le répertoire `lib` de votre installation UIMA. Le cc_cas2jdbc.jar est déjà dans le répertoire `lib` de votre installation UIMA.
8. Créer le fichier de mappage de la structure d'analyse commune à la base de données qui définit quels résultats d'analyse de texte doivent être stockés dans une base de données. Vous pouvez utiliser le fichier de mappage `sampleMapping.xml` de *UIMA_SDK_INSTALL*`/docs/examples/descriptors/cas_consumer` comme exemple pour créer votre propre fichier de mappage.

   Utilisez le fichier de schéma XML appelé `CasToJDBCMapping.xsd` de *UIMA_SDK_INSTALL*`/docs/examples/descriptors/cas_consumer` pour valider le fichier de mappage de la structure d'analyse commune à la base de données. Pour des raisons de performances, le consommateur de la structure d'analyse commune à la base de données ne valide pas le fichier de mappage, vous devez faire cela vous-même.

La procédure d'exécution du consommateur dans l'architectureUIMA est décrite dans la documentation UIMA.

L'exemple suivant montre comment les paramètres obligatoires doivent être définis dans le descripteur :

```
 ...
 <nameValuePair>
  <name>mappingFile</name>
  <value>
   <string>D:/temp/MyMapping.xml</string>
  </value>
 </nameValuePair>
<nameValuePair>
  <name>docMetadata_Type</name>
  <value>
   <string>uima.tcas.DocumentAnnotation</string>
  </value>
 </nameValuePair>
<nameValuePair>
  <name>docId_Feature</name>
  <value>
   <string>end</string>
  </value>
 </nameValuePair>

...
```

Le tableau présente les paramètres de configuration dans l'ordre dans lequel ils apparaissent dans le fichier descripteur et indique lesquels sont obligatoires :

Tableau 1. Les paramètres de configuration du fichier descripteur du consommateur de la structure d'analyse commune à la base de données

Paramètre	Description	Obligatoire
mappingFile	Le chemin d'accès absolu au fichier de mappage de la structure d'analyse commune à la base de données, par exemple, `D:/temp/sample.xml`. Sur les systèmes Windows, utilisez "/" en tant que séparateur de chemin d'accès.	true
encryptionClass	Ne définissez pas ce paramètre, il est uniquement utilisé dans la recherche d'entreprise pour permettre au consommateur de la structure d'analyse commune à la base de données de fonctionner avec les fichiers de mappage chiffrés.	false
docMetadata_Type	Le type UIMA à partir duquel toutes les métadonnées pour les fonctions intégrées sont extraites.	true
docId_Feature	La fonction ou le chemin de fonctions du type de métadonnées à partir duquel l'ID numérique du document est extrait. Il doit être de type nombre entier et est requis pour toutes les fonctions intégrées qui nécessitent l'ID, telles que `uniqeId()`, `objectId()` et `fsId()`.	true
docUri_Feature	La fonction ou le chemin de fonctions du type de métadonnées à partir duquel l'URI du document est extrait. Il doit être de type chaîne.	false
IsCompleted_Feature	La fonction ou le chemin de fonctions du type de métadonnées qui indique si le document actuel est morcelé à travers plusieurs structures d'analyse communes.	false
chunkNumber_Feature	La fonction ou le chemin de fonctions du type de métadonnées qui indique le numéro ultérieur du morceau en cours.	false

Utilisation de l'annotateur d'expressions régulières dans l'architecture UIMA

Utilisez l'annotateur d'expressions régulières pour détecter des entités ou unités d'informations dans un document texte. Vous pouvez personnaliser l'annotateur pour votre domaine pour satisfaire vos besoins en termes de recherche.

Pour exécuter l'exemple d'annotateur d'expressions régulières qui détecte les numéros de téléphone, les URL et les adresses électroniques ou pour utiliser l'exemple d'annotateur en tant que base pour créer votre propre version personnalisée de l'annotateur d'expressions régulières dans votre environnement UIMA, vous devez avoir :

1. Le descripteur de l'annotateur d'expressions régulières dans le répertoire *UIMA_SDK_INSTALL*/docs/examples/descriptors/analysis_engine.
2. L'exemple de jeu de règles et la description du système de types dans le répertoire *UIMA_SDK_INSTALL*/docs/examples/regex.
3. Un exemple de fichier texte auquel l'exemple de jeu de règles peut s'appliquer dans le répertoire *UIMA_SDK_INSTALL*/docs/data, appelé of_sample_regex.txt.

La procédure d'exécution de l'annotateur dans l'architecture UIMA est décrite dans la documentation UIMA.

Affichage des résultats de l'annotateur de base et de l'analyse de texte personnalisée

Pour consulter les résultats d'analyse produits après l'analyse syntaxique et par tous les annotateurs de la recherche d'entreprise, vous devez mettre à jour les propriétés des collections de documents pour produire une version XML lisible des résultats d'analyse stockés dans la structure d'analyse commune.

A propos de cette tâche

Vous utilisez la sérialisation XML des résultats d'analyse des annotateurs stockés dans la structure d'analyse commune pour :

- Consulter les résultats après analyse syntaxique, avant que les annotateurs de base ne soient traités.
- Consulter les résultats après analyse syntaxique et marquage sémantique (en exécutant les annotateurs de base de la recherche d'entreprise). Cela peut vous aider à déterminer les structures de données d'entrée des analyses personnalisées que vous voulez développer et qui s'exécuteront toujours après les annotateurs de base.
- Consulter et valider les résultats d'une analyse personnalisée exécutée sur une collection de documents plus petite de la recherche d'entreprise à des fins de test avant de décider de l'exécuter sur une collection complète.

La sérialisation XML produit deux jeux de résultats :

- Les résultats après analyse syntaxique. Ils incluent les mappages de zones et les métadonnées de documents.
- Les résultats après analyse syntaxique et marquage sémantique et, si elle est sélectionnée, l'analyse de texte personnalisée. Ils incluent toutes les annotations et tous les marqueurs sémantiques produits.

Procédure

Pour produire une version XML lisible des résultats d'analyse, procédez comme suit :

1. Ouvrez le fichier collection.properties de *ES_NODE_ROOT*/master_config/<CollectionID>.parserdriver avant de commencer à analyser les documents de votre collection.
2. Pour consulter les résultats après analyse syntaxique, ajoutez la ligne suivante au fichier collection.properties : trevi.parser.dumpXCas=<your_dump_directory>

   Votre répertoire de clichés doit déjà exister.

   a. Sélectionnez le type de sortie que vous voulez. La sortie inclut toujours la description du système de types utilisée pour les résultats d'analyse syntaxique, appelée OmniFindParserTypeSystem.xml. Ajoutez l'une des lignes suivantes :

      - Pour afficher la sortie des 25 derniers fichiers traités, ajoutez trevi.parser.maxXCasFileCount=25.

        Vous pouvez déterminer le nombre de fichiers vous-même, mais nous vous conseillons de ne pas le définir trop haut.

        N'oubliez pas que la mémoire tampon de sortie du fichier est sans cesse remplacée une fois sa taille maximale atteinte. Cela signifie également que le document avec le nombre le plus élevé n'a pas besoin d'être le dernier traité.

        La sortie inclut les fichiers suivants : OmniFindParserXCasDump1.xml suivi d'OmniFindParserXCasDump2.xml, etc. jusqu'à ce que 25 fichiers soient répertoriés.

      - Pour afficher la sortie de documents spécifiques, ajoutez l'URI de document trevi.parser.xCasURI.1=file://home/test/file1.txt.

        Vous pouvez ajouter un certain nombre de documents, toutefois, ils doivent être numérotés dans l'ordre croissant en commençant par 1, sans oublier de nombre. Par exemple, le deuxième document serait trevi.parser.xCasURI.2=file://home/test/file2.txt et le troisième trevi.parser.xCasURI.3=file://home/test/file3.txt.

        La sortie inclut les fichiers suivants : OmniFindParserXCasDumpURI_1.xml, OmniFindParserXCasDumpURI_2.xml, etc. pour autant de noms de fichiers qu'il en est répertorié
3. Pour consulter les résultats après marquage sémantique, ajoutez la ligne suivante : trevi.tokenizer.dumpXCas=<your_dump_directory>

   Là encore, votre répertoire de clichés doit déjà exister.

   a. Sélectionnez le type de sortie que vous voulez. La sortie créée inclut également toujours la description du système de types utilisée pour le marquage sémantique et les résultats d'analyse syntaxique, appelée OmniFindTypeSystem.xml. Ajoutez l'une des lignes suivantes :

      - Pour afficher la sortie de 25 derniers fichiers traités, ajoutez trevi.tokenizer.maxXCasFileCount=25.

        Vous pouvez déterminer le nombre de fichiers vous-même, mais nous vous conseillons de ne pas le définir trop haut.

        N'oubliez pas que la mémoire tampon de sortie du fichier est sans cesse remplacée une fois sa taille maximale atteinte. Cela signifie également que le document avec le nombre le plus élevé n'a pas besoin d'être le dernier traité.

La sortie inclut les fichiers suivants : OmniFindXCasDump1.xml, OmniFindXCasDump2.xml, etc. jusqu'à ce que 25 fichiers soient répertoriés.

- Pour afficher la sortie de documents spécifiques, ajoutez l'URI de document `trevi.tokenizer.xCasURI.1=file://home/test/file1.txt`.

  Vous pouvez ajouter un certain nombre de documents, toutefois, ils doivent être numérotés dans l'ordre croissant en commençant par 1, sans oublier de nombre. Par exemple, le deuxième document serait `trevi.tokenizer.xCasURI.2=file://home/test/file2.txt` et le troisième `trevi.tokenizer.xCasURI.3=file://home/test/file3.txt`.

  La sortie inclut les fichiers suivants : OmniFindXCasDumpURI_1.xml, OmniFindXCasDumpURI_2.xml, etc. pour autant de noms de fichiers qu'il en est répertorié.

Dans la recherche d'entreprise, vous pouvez utiliser le visualiseur d'annotations XCAS pour afficher le contenu des fichiers XML. Démarrez le visualiseur d'annotations XCAS en exécutant le fichier script `xcasAnnotationViewer` situé dans le répertoire `ES_INSTALL_ROOT/bin`. Vous êtes invité à fournir :

- Votre répertoire de clichés dans lequel les résultats sont placés après analyse syntaxique ou marquage sémantique.
- Le fichier descripteur, soit OmniFindParserTypeSystem.xml (pour les résultats de l'analyseur syntaxique), soit OmniFindTypeSystem.xml (pour les résultats de marquage sémantique et d'analyse), également dans votre répertoire de clichés.

Le fait de sélectionner un document dans la liste affiche les résultats d'analyse pour celui-ci. Le fait de cliquer sur une annotation mise en évidence dans le document affiche les détails de celle-ci.

Description du système de types

Le système de types définit les types des objets et leurs propriétés (ou fonctions), qui peuvent être instanciées dans une structure d'analyse commune.

Chaque moteur d'analyse dispose de ses propres descriptions de systèmes de types décrivant les éléments d'entrée requis et les types de sortie pour les annotateurs du moteur d'analyse. Les descriptions de systèmes de types sont propres au domaine d'application.

Les systèmes de types incluent les définitions de types, leurs propriétés et la hiérarchie à héritage simple des types. Une structure d'analyse commune doit être conforme à un système de types particulier.

Les fonctions et les types définis dans la description du système de types doivent également être utilisés dans tous les fichiers de mappage associés à l'analyse de documents, y compris le fichier de mappage des éléments XML à la structure d'analyse commune, le fichier de mappage de la structure d'analyse commune à l'index et le fichier de mappage de la structure d'analyse commune à la base de données.

La description du système de types d'un annotateur peut faire partie du descripteur de l'annotateur ou elle peut faire partie d'un fichier séparé du descripteur du type de système. Elle fait parfois partie du descripteur d'un autre annotateur contenu dans le même moteur d'analyse.

Lorsque vous avez fini de développer et de tester votre moteur d'analyse dans votre environnement UIMA, le fichier archive (fichier .pear file) que vous créez et téléchargez sur la recherche d'entreprise contient vos fichiers de logique d'analyse ainsi que votre description du système de types.

Les annotateurs de base de la recherche d'entreprise utilisent trois descriptions de systèmes de types ; une description de système de types principale qui est toujours incluse et deux autres que vous pouvez activer éventuellement pour faire passer le traitement d'analyse de base des collections de documents sur le mode d'analyse avancée. Le fait d'avoir à inclure une seule ou les deux descriptions de systèmes de types étendues dépend des résultats du traitement d'analyse du texte supplémentaire que vous voulez inclure durant le traitement d'analyse de base.

Vous pouvez activer le mode d'analyse avancée en incluant un seul ou les deux systèmes de types d'extension. En mode d'analyse avancée, d'autres fonctions d'analyse sont disponibles durant le traitement d'analyse de base et sont enregistrées vers la structure d'analyse commune. Par exemple, si vous avez besoin de davantage d'informations sur un marqueur sémantique (plus d'informations de fonctions), notamment si vous voulez connaître tous les lemmes possibles pour le marqueur sémantique ou si le lemme est un mot vide ou la classe de mots du lemme ou des fonctions spéciales pour le traitement morphologique, également pour le japonais, vous devez activer le mode d'analyse avancée.

Tâches associées

«Passage du mode d'analyse de base au mode d'analyse avancé»

Référence associée

«Types et fonctions définis pour la recherche d'entreprise», à la page 17

Passage du mode d'analyse de base au mode d'analyse avancé

Pour faire passer le traitement des collections de documents mené par les annotateurs de base de la recherche d'entreprise du mode d'analyse de base au mode d'analyse avancé, vous devez inclure les descriptions de système de types pour le mode d'analyse avancé.

Restrictions

Il existe deux descriptions de système de types que vous pouvez sélectionner pour activer le mode d'analyse avancé :

- La description `tt_extension_typesystem`, qui inclut des informations de fonctions de type lexical plus détaillées sur les lemmes.
- La description `dlt_extension_typesystem`, qui inclut des fonctions morphologiques supplémentaires et des types lexicaux spéciaux.

Procédure

Pour passer du traitement de collections de base au mode d'analyse avancé, procédez comme suit :

1. Ouvrez le fichier tt_core_typesystem.xml du répertoire *ES_NODE_ROOT*/`master_config/`*CollectionID*`.parserdriver/specifiers`. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix.

2. Supprimez les codes de commentaires entourant l'élément <import> de la section <imports> pour inclure l'un des fichiers de description de système de types d'extension ou les deux.

```
<imports>
<!-- importe le tt_extension_typsystem pour l'analyse avancée -->
<!-- <import location="tt_extension_typesystem.xml"/>-->
<!-- importe le typesystem d'extension dlt -->
<!-- <import location="dlt_extension_typesystem.xml"/> -->
</imports>
```

3. Ouvrez les deux fichiers descripteurs jfrost.xml et jfrost_ngram.xml et modifiez le contenu de l'élément <outputs> pour inclure les types (dans un élément <type>) et les fonctions (dans un élément <feature>) répertoriés dans l'élément <description> de la section <capabilities> que vous voulez inclure durant l'analyse. Enregistrez vos modifications.
4. Ouvrez le fichier descripteur jtok.xml et modifiez le contenu de l'élément <outputs> pour inclure les fonctions (dans un élément <feature>) répertoriées dans l'élément <description> de la section <capabilities> que vous voulez inclure durant l'analyse. Enregistrez vos modifications.
5. Ouvrez le fichier descripteur es_tok_no_stw.xml et ici aussi, modifiez le contenu de l'élément <outputs> pour inclure les fonctions (dans un élément <feature>) répertoriées dans l'élément <description> de la section <capabilities> que vous voulez inclure durant l'analyse. Enregistrez vos modifications.
6. Lorsque vous passez en mode d'analyse avancé, vous devez réanalyser votre collection de documents.

Concepts associés

«Description du système de types», à la page 15

Référence associée

«Types et fonctions définis pour la recherche d'entreprise»

Types et fonctions définis pour la recherche d'entreprise

Le système type défini pour la recherche d'entreprise couvre la gestion des métadonnées de document et l'analyse linguistique de base.

Les types utilisés dans la recherche d'entreprise sont définis dans trois fichiers de description de systèmes de types distincts, en commençant par le fichier de description de systèmes de types contenant les types principaux, toujours requis pour toutes les analyses linguistiques de base, puis en poursuivant par les descriptions des systèmes de types définissant les fonctions linguistiques avancées, normalement uniquement requises en mode d'analyse avancé.

L'analyse linguistique de base sous la forme de reconnaissance de la langue du document et de segmentation est toujours effectuée lors de l'indexation d'un document, que l'analyse personnalisée soit ou non sélectionnée. Lors de l'analyse de documents de base, la description tt_core_typesystem est utilisée et les informations suivantes sont ajoutées à la structure d'analyse commune que vous pouvez utiliser dans l'analyse personnalisée ultérieure :

- Métadonnées de document de type com.ibm.es.tt.DocumentMetaData.
- Informations de structure de document telles que phrase et annotations de paragraphe de type uima.tt.SentenceAnnotation et uima.tt.ParagraphAnnotation.
- Annotations lexicales telles que marqueurs sémantiques et composés de type uima.tt.TokenAnnotation.

La description tt_core_typesystem est appropriée pour la plupart du traitement d'analyse de texte.

Si vous voulez faire passer le traitement des collections en mode d'analyse avancée, vous pouvez inclure les deux systèmes de types suivants. Les systèmes de types incluent premièrement des fonctions supplémentaires qui ne sont pas créées durant le traitement linguistique de base.

- tt_extension_typesystem qui inclut davantage d'informations sur les marqueurs sémantiques, lemmes, paragraphes et phrases
- dlt_core_typesystem qui contient certains des types d'annotations étendus IBM LanguageWare, par exemple, les URL et les adresses. Il inclut également des fonctions morphologiques qui ne sont pas utilisées fréquemment.

tt_core_typesystem

Les fonctions et types suivants sont définis dans la description tt_core_typesystem:

uima.tcas.DocumentAnnotation
L'annotation de document contient les métadonnées de document et a la fonction suivante :

- categories avec les catégories de document ajoutées à un catégoriseur de texte. Chaque catégorie ajoutée est de type com.tt.CategoryConfidencePair
- languageCandidates avec les langues de document automatiquement détectées durant l'analyse syntaxique. Les langues sont ajoutées à une liste de type com.tt.LanguageConfidencePair, avec la langue la plus pertinente répertoriée en premier
- id avec l'ID document, tel que l'URL

uima.tt.TTAnnotation
Il s'agit du type principal pour les annotations définies dans tt_core_typesystem. Le super-type est uima.tcase.Annotation. Il dispose des types suivants :

uima.tt.DocStructureAnnotation
Annotations relatives à la structure de document. Elle présente les sous-types suivants :

uima.tt.SentenceAnnotation
Phrases

uima.tt.ParagraphAnnotation
Paragraphe de document

uima.tt.LexicalAnnotation
Annotations lexicales telles que marqueurs sémantiques ou expressions contenant plusieurs mots. Elle présente les sous-types suivants :

uima.tt.TokenLikeAnnotation
Annotations de marqueurs sémantiques uniques pouvant avoir les fonctions suivantes :

- tokenProperties avec les propriétés des marqueurs sémantiques
- lemma avec le lemme ou radical du terme

- `normalizedCoveredText` avec la représentation normalisée du texte couvert

Ce type d'annotation présente les sous-types suivants :

uima.tt.TokenAnnotation
Marqueurs sémantiques réels à distinguer des parties composées.

uima.tt.CompPartAnnotation
Les parties composées d'un terme.

uima.tt.CompoundAnnotation
L'annotation d'un marqueur sémantique composé. Le marqueur sémantique composé s'étend généralement plus qu'une annotation de marqueur sémantique.

uima.tt.MultiTokenAnnotation
Annotation lexicale comprenant plusieurs marqueurs sémantiques. Ce type d'annotation présente les sous-types suivants :

uima.tt.StopwordAnnotation
Annotations de mots vides. Les mots vides peuvent également être des mots de plusieurs termes.

uima.tt.SynonymAnnotation
L'annotation d'un terme pour lequel il existe des synonymes. Elle a la fonction `synonyms` qui répertorie les synonymes trouvés pour le terme.

uima.tt.SpellCorrectionAnnotation
L'annotation d'un terme pour lequel il existe des corrections orthographiques. Elle a la fonction `correctionTerms` qui répertorie les corrections possibles dans un ordre trié, commençant par les corrections les plus pertinentes.

uima.tt.MultiWordAnnotation
L'annotation d'un terme composé de plusieurs mots.

uima.CAS.TOP
Elément principal du système de types. Elle présente les sous-types suivants :

uima.tt.KeyStringEntry
Le type abstrait pour les structures de données String. Il inclut la fonction `key` qui contient la clé de chaîne et le sous-type suivant :

uima.tt.Lemma
Entrées de lemmes de dictionnaire.

uima.tt.CategoryConfidencePair
La valeur de confiance pour la catégorie trouvée. Elle a les fonctions suivantes :

- `categoryString` avec le nom de la catégorie
- `categoryConfidence` avec la valeur de confiance pour la catégorie
- `mostSpecific` avec un indicateur signalant si cette catégorie est la plus spécifique au document

- `taxonomy` avec le nom de la taxinomie de laquelle la catégorie est dérivée

uima.tt.LanguageConfidencePair
La valeur de confiance pour la catégorie trouvée. Ce type inclut les fonctions `languageConfidence`, `language` et `languageID`.

tt_extension_typesystem

Le `tt_extension_typesystem` inclut des fonctions d'analyse de texte supplémentaires pour un traitement plus avancé.

uima.tt.TokenLikeAnnotation
Ce type d'annotation dans le `tt_extension_typesystem` a les fonctions suivantes :

- `lemmaEntries` répertorie tous les lemmes possibles pour le marqueur sémantique. Les éléments de la liste sont de type `uima.tt.Lemma`
- `tokenNumber`
- `stopwordToken`

uima.tt.Lemma
Cette annotation de type `uima.tt.KeyStringEntry` a les fonctions suivantes :

- `isStopword` est vrai si le lemme est un mot vide
- `isDeterminer` est vrai si le lemme est un déterminateur
- `partOfSpeech`. Les codes de description numéraux de la classe des mots suivants existent :
  - 0 : inconnu
  - 1 : pronom
  - 2 : verbe
  - 3 : nom
  - 4 : adjectif
  - 5 : adverbe
  - 6 : adposition
  - 7 : interjection
  - 8 : conjonction

uima.tt.DocStructureAnnotation
Annotations relatives à la structure de document. Celle-ci présente les sous-types suivants :

uima.tt.SentenceAnnotation
Phrase du document. Elle a la fonction `sentenceNumber`.

uima.tt.ParagraphAnnotation
Paragraphe du document. Elle a la fonction `paragraphNumber`.

dlt_extension_typesystem

Le `dlt_extension_typesystem` inclut des fonctions supplémentaires utilisées par IBM LanguageWare.

uima.tt.LexicalAnnotation
Cette annotation présente les sous-types suivants :

uima.tt.TokenLikeAnnotation
Dans le `dlt_extension_typesystem`, cette annotation a les fonctions suivantes :

- `synonymEntries`
- `frost_TokenType`
- `inflectedForms`
- `spellAid`
- `decomposition`

com.ibm.dlt.uimatypes.FilePath

com.ibm.dlt.uimatypes.Email

com.ibm.dlt.uimatypes.Number

com.ibm.dlt.uimatypes.URL

com.ibm.dlt.uimatypes.Date

com.ibm.dlt.uimatypes.Time

com.ibm.dlt.uimatypes.Tel

com.ibm.dlt.uimatypes.Currency

com.ibm.dlt.uimatypes.Acronym

uima.tt.TokenLikeAnnotation
Ce type d'annotation du `dlt_extension_typesystem` a le type suivant :

com.ibm.dlt.uimatypes.MWU
Ce type est utilisé par IBM LanguageWare pour annoter les expressions composées de plusieurs mots.

uima.tt.KeyStringEntry
Annotations de chaîne. Celle-ci présente les sous-types suivants :

uima.tt.Lemma
Elle a les fonctions suivantes :

- `frost_Constraints` avec indicateurs de contrainte
- `frost_MorphBitMasks` contenant un tableau de masques de contrôle des données morphologiques
- `frost_ExtendedPOS` avec informations de classes de mots étendues, telles que JPOS pour le japonais et CPOS pour le chinois
- `frost_JKom` contenant des données morphologiques japonaises
- `frost_JPStart` contenant des données d'analyse de début japonaises
- `morphID` contenant des propriétés de lemmes

uima.tcas.Annotation
Celle-ci présente le sous-type suivant :

com.ibm.dlt.uimatypes.Decomp_Analysis
Analyse structurelle complète d'un composé. Celle-ci a les fonctions suivantes :

- `headComponentIndex` avec le composant de tête du composé
- `route` contenant une liste des marqueurs sémantiques comprenant une route de décomposition unique

Référence associée

«Exemple de description de système de types», à la page 25

Types et fonctions spécifiques pour la recherche d'entreprise

Les types et fonctions définis dans la description of_typesystem couvrent des types spécifiques pour OmniFind Enterprise Edition. Ces types sont utilisés pour des métadonnées spécifiques aux documents. Ils décrivent également la représentation des zones et des informations de marquage XML ou des ancres HTML.

La description of_typesystem n'est pas définie dans le SDK (Software Development Kit) UIMA. Si vous souhaitez utiliser un de ces types lors de la création d'un annotateur dans l'architecture UIMA, vous devez définir une nouvelle fois les types dans la description du système type du moteur d'analyse. Vous pouvez, par exemple, accéder à des informations sur la sécurité des documents ou accéder au type de moteur de balayage ou de document.

Les fonctions et types suivants sont définis dans la description of_typesystem :

uima.tcas.DocumentAnnotation
L'annotation de document UIMA standard est développée par la fonction suivante :

esDocumentMetaData
Contient des métadonnées de document du type `com.ibm.es.tt.DocumentMetaData`.

com.ibm.es.tt.DocumentMetaData
Le type de métadonnées de document a les fonctions suivantes. Les fonctions sont connectées à la fonction d'annotation de document `esDocumentMetaData`.

crawlerId
Nom du moteur de balayage. La valeur de la fonction est de type `uima.cas.String`.

dataSource
Un des types de source de données suivants. La valeur de la fonction est de type `uima.cas.String`.

- CM, pour les documents explorés par le moteur de balayage DB2 Content Manager.
- Database, pour les documents explorés par le moteur de balayage Base de données JDBC.
- DB2, pour les documents explorés par le moteur de balayage DB2.
- DominoDoc, pour les documents explorés par le moteur de balayage Domino Document Manager.
- Exchange, pour les documents explorés par le moteur de balayage Exchange Server.
- NNTP, pour les documents explorés par le moteur de balayage NNTP.
- Notes, pour les documents explorés par le moteur de balayage Notes.
- QuickPlace, pour les documents explorés par le moteur de balayage QuickPlace.
- Seedlist, pour les documents explorés par le moteur de balayage Liste de valeurs de départ.

- UnixFS, pour les documents explorés par le moteur de balayage Système de fichiers UNIX.
- VBR, pour les documents explorés par le moteur de balayage Content Edition.
- WCM, pour les documents explorés par le moteur de balayage Web Content Management.
- Web, pour les documents explorés par le moteur de balayage Web.
- WinFS, pour les documents explorés par le moteur de balayage Système de fichiers Windows.
- WP, pour les documents explorés par le moteur de balayage WebSphere Portal.

dataSourceName
Nom du moteur de balayage (source de données). La valeur de la fonction est de type `uima.cas.String`.

docType
Un des types de document suivants. La valeur de la fonction est de type `uima.cas.String`.

- text/html
- application/postscript
- application/pdf
- application/x-mspowerpoint
- application/msword
- application/x-msexcel
- application/rtf
- application/vnd.lotus-wordpro
- application/x-lotus-123
- application/vnd.lotus-freelance
- text/xml
- text/plain
- application/x-js-taro (Ichitaro)

securityTokens
Jetons de sécurité du document. La valeur de la fonction est de type `uima.cas.StringArray`.

date Date du document. La valeur de la fonction est de type `uima.cas.String`.

baseUri
URI de base de la page. La valeur de la fonction est de type `uima.cas.String`.

metaDataFields
La valeur de la fonction est de type `uima.cas.FSArray`. Chaque élément de ce tableau est de type `com.ibm.es.tt.MetaDataField`.

redirectUrl
URL redirigée. La valeur de la fonction est de type `uima.cas.String`.

mimeType
Type de MIME ou document, par exemple XML. La valeur de la fonction est de type `uima.cas.String`.

url URL du document. La valeur de la fonction est de type `uima.cas.String`.

com.ibm.es.tt.CommonFieldParameters
Les paramètres de zone commune incluent :

searchable
Indicateur définissant s'il est possible d'effectuer une recherche sans texte dans la zone.

fieldSearchable
Indicateur définissant s'il est possible d'effectuer une recherche dans la zone.

parametric
Indicateur définissant s'il est possible d'effectuer une recherche dans la zone, via une requête paramétrique.

showInSearchResult
Indicateur définissant si les données annotées sont incluses dans les détails des résultats de la recherche.

resolveConflict
Indicateur permettant de résoudre les conflits de métadonnées entre `MetadataPreferred`, `ContentPreferred` et `Coexist`. La valeur de la fonction est de type `uima.cas.String`.

name Nom de la zone. Vous pouvez rechercher cette zone en indiquant son nom. La valeur de la fonction est de type `uima.cas.String`.

sortable
Indicateur définissant s'il est possible de trier les chaînes dans la zone.

exactMatch
Indicateur signalant que la recherche doit correspondre exactement aux termes de la requête.

com.ibm.es.tt.ContentField
L'annotation de zone de contenu a la fonction suivante :

parameters
Les paramètres de zone de contenu sont de type `com.ibm.es.tt.CommonFieldParameters`.

com.ibm.es.tt.MetaDataField
Les données de la zone de métadonnées ne font pas partie du contenu du document mais elles sont stockées dans la fonction "text" :

parameters
Paramètres de la zone de métadonnées de type `com.ibm.es.tt.CommonFieldParameters`.

text Le texte des métadonnées est stocké dans cette fonction de type `uima.cas.String`.

com.ibm.es.tt.Anchor
Annotation d'ancrage pour le texte d'ancrage dans les documents HTML. Elle dispose de la fonction suivante :

uri URI cible du texte d'ancrage. La valeur de la fonction est de type `uima.cas.String`.

com.ibm.es.tt.MarkupTag
Annotations d'informations de marquage, par exemple, d'une balise XML. Les informations de marquage sont stockées dans les fonctions suivantes :

name Nom de la balise de marquage. La valeur de la fonction est de type `uima.cas.String`.

depth Profondeur d'imbrication. La valeur de la fonction est de type `uima.cas.Integer`.

attributeName
Nom de l'attribut de la fonction. La valeur de la fonction est de type `uima.cas.StringArray`.

attributeValues
Chaîne de valeurs pour l'attribut. La valeur de la fonction est de type `uima.cas.StringArray`.

Exemple de description de système de types

Décrit les structures de fonctions (structures de données sous-jacentes qui représentent les résultats de l'analyse) utilisées dans l'analyse personnalisée.

La description du système de types doit faire partie de l'archive du moteur de balayage (fichier .pear) importé à partir de l'environnement UIMA dans la recherche d'entreprise.

L'exemple de description de système de types suivant décrit des rapports de police qui contiennent des informations sur les suspects, le lieu du crime, l'heure du crime et la date :

La même description du système de types exemple est utilisée dans l'ensemble des sections d'analyse de texte qui décrivent les différents types de mappage que vous pouvez sélectionner avec l'analyse personnalisée.

```
<?xml version="1.0" encoding="UTF-8"?>
<typeSystemDescription>
   <name>Système de types de rapport de police</name>
   <description>Description de système de types pour
        les rapports de police</description>
   <version>1.0</version>
   <types>
     <typeDescription>
       <name>com.ibm.omnifind.types.PoliceReport</name>
       <description>Annote un rapport de police</description>
       <supertypeName>uima.tcas.Annotation</supertypeName>
       <features>
         <featureDescription>
           <name>time</name>
           <description>Heure rapportée pour le moment du crime
              </description>
           <rangeTypeName>com.ibm.omnifind.types.Time</rangeTypeName>
         </featureDescription>
         <featureDescription>
           <name>date</name>
           <description>Date du crime</description>
           <rangeTypeName>com.ibm.omnifind.types.Date</rangeTypeName>
         </featureDescription>
         <featureDescription>
           <name>location</name>
           <description>Endroit où s'est déroulé le crime</description>
```

```
      <rangeTypeName>com.ibm.omnifind.types.City</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>knownSuspects</name>
      <description>Contient des annotations de type Suspect</description>
      <rangeTypeName>uima.cas.FSArray</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>crimeDescription</name>
      <description>Brève description du crime</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.City</name>
  <description>Nom de ville</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>cityName</name>
      <description>Nom de la ville</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>cityDistrict</name>
      <description>Nom du district</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Person</name>
  <description>Annotation de la personne</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>role</name>
      <description>Par exemple, suspect ou témoin</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>firstName</name>
      <description>Prénom de la personne</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>surName</name>
      <description>Nom de famille de la personne</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>title</name>
      <description>Par exemple, M. ou Mme</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>gender</name>
      <description>Homme ou femme</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Suspect</name>
  <description>Suspect trouvé</description>
  <supertypeName>com.ibm.omnifind.types.Person</supertypeName>
```

```
  <features>
    <featureDescription>
      <name>description</name>
      <description>Description du suspect ;
    par exemple, barbu avec des lunettes noires</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Date</name>
  <description>Date</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>year</name>
      <description>L'année ; par exemple, 2005</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>month</name>
      <description>Mois en chiffres ; par exemple, 7</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>day</name>
      <description>Jour en chiffres</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>dayOfWeek</name>
      <description>Le jour de la semaine ; par exemple lundi</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>quarter</name>
      <description>Le trimestre ; par exemple T1-2005</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>englDate</name>
      <description>Date au format mm/jj/aaaa</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
  </features>
</typeDescription>
<typeDescription>
  <name>com.ibm.omnifind.types.Time</name>
  <description>Heure</description>
  <supertypeName>uima.tcas.Annotation</supertypeName>
  <features>
    <featureDescription>
      <name>hours</name>
      <description>Heures de 0h à 23h</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>minutes</name>
      <description>Minutes dans l'heure</description>
      <rangeTypeName>uima.cas.Integer</rangeTypeName>
    </featureDescription>
    <featureDescription>
      <name>timeOfDay</name>
      <description>Périodes telles que le matin, le midi</description>
      <rangeTypeName>uima.cas.String</rangeTypeName>
    </featureDescription>
```

```
      </features>
    </typeDescription>
  </types>
</typeSystemDescription>
```

Marquage XML dans l'analyse et la recherche

Vous pouvez mapper les informations des structures XML se trouvant dans un document directement à une structure d'analyse commune sans l'aide d'un annotateur UIMA.

Si les documents de votre collection sont au format XML et que vous souhaitez exploiter le marquage XML lors de l'analyse de texte ou de la recherche sémantique, vous disposez des options suivantes :

Recherche XML native

Employez cette option si vous souhaitez utiliser l'ensemble des balises et attributs XML, tels qu'ils apparaissent dans le document lors de la recherche sémantique. Par exemple, si vous avez des documents de facturation qui contiennent un élément `<addressee>`, l'activation de la recherche XML native permet d'utiliser cette balise dans une requête de recherche sémantique pour un nom de client particulier dans cet élément.

Avec cette option, la structure XML du document est représentée dans la structure d'analyse commune à l'aide du type `com.ibm.es.tt.MarkupTag`. Pour chaque balise XML, une annotation de ce type est créée. Cette annotation contient le nom de la balise, ses attributs et le contenu de l'attribut. Ces informations sont toujours indexées et sont accessibles pour la recherche sémantique.

La recherche XML native ne requiert aucun fichier de configuration de mappage. Vous pouvez activer la recherche XML native à partir de la console d'administration pour la recherche d'entreprise.

Mappage des éléments XML à la structure d'analyse commune

Utilisez cette option dans les situations suivantes :

- La sémantique de certains éléments XML est précise et peut être utilisée pour d'autres étapes de l'analyse de texte. Les étapes de l'analyse peuvent être effectuées directement sur les annotations et les fonctions créées à partir des structures XML et sont dérivées des différents formats potentiels des documents d'origine. Par exemple, l'élément `<addressee>` des documents concernant la facturation contient généralement des noms de client. Lors de l'utilisation du mappage des éléments XML à la structure d'analyse commune, le contenu de cet élément peut être mappé directement aux annotations de type `Customer`. Un annotateur peut alors déduire une relation client-situé-à, à l'aide des informations entourant l'annotation `Customer`.
- Vous souhaitez limiter la portée du traitement d'un annotateur personnalisé à des zones spécifiées dans l'entrée XML. Par exemple, vous pouvez limiter l'analyse au contenu des balises `<technicianComment>` uniquement dans un annotateur qui détecte les problèmes automobiles.
- Vous souhaitez restreindre le traitement d'analyse de texte ainsi que les recherches suivantes à certaines parties du document XML et supprimer le contenu non textuel ou inapproprié.

- Vous souhaitez mapper des balises XML ayant des noms différents à une étendue commune à utiliser dans la recherche sémantique. Par exemple, mapper <mainHeading> ou <doc> au titre.

Dans ces cas, vous devez créer un fichier de mappage des éléments XML à la structure d'analyse commune qui définit quels éléments XML mappent quelles structures de fonctions. Les structures de fonctions que vous définissez dans le fichier de mappage sont créées lors de l'analyse des documents et il est possible d'y accéder à l'aide des annotateurs personnalisés.

Vous pouvez utiliser plusieurs fichiers de mappage des éléments XML à la structure d'analyse commune pour une collection de documents. L'élément <identifier> permet de déterminer quel fichier de mappage est utilisé pour quel document XML. L'élément <identifier> du fichier de mappage doit correspondre à l'élément principal du document XML. Par exemple, si l'élément principal du document est doc, la valeur de l'élément <identifier> du fichier de mapping doit également être "doc".

Si aucune correspondance n'est trouvée, le programme recherche un fichier de mappage avec l'élément <identifier> ayant la valeur par défaut. Si aucun mappage par défaut n'est trouvé, les sections de texte du document (sans informations de balise) sont mappées à l'annotation de document dans la structure d'analyse commune.

Si vous souhaitez extraire des informations qui se trouvent uniquement dans les parties pertinentes d'un document, tout en ignorant les parties inappropriées, il vous suffit d'indiquer quels éléments XML des documents contiennent les informations appropriées. Cette opération est appelée extraction. Par exemple, vous pouvez extraire les informations se trouvant dans les sections title et body tout en ignorant les informations des sections author, date, ID et publisher.

L'extraction du contenu peut améliorer le traitement de l'analyse pour les types de document XML suivants :

- Les documents incluant de grandes quantités de contenu non soumis à l'analyse, des pièces jointes binaires, par exemple. L'utilisation de l'extraction de contenu réduit de manière significative la taille du document, ce qui fait que le traitement est plus rapide et les erreurs d'analyse dues au fait que des données inappropriées sont utilisées sont évitées.
- Les documents contenant du texte inapproprié, par exemple, des documents contenant des informations éditoriales dans des balises <note>. Le fait d'ignorer ces informations génère un meilleur résultat lors de l'analyse du contenu du document.

L'utilisation de la recherche XML native et des options d'extraction de contenu dans le mappage des éléments XML à la structure d'analyse commune sont des options qui s'excluent mutuellement car soit l'ensemble du contenu, soit uniquement le contenu spécifié peut être pris en considération. Si vous indiquez l'extraction de contenu, le mappage XML natif est ignoré. Sans l'extraction de contenu, vous pouvez avoir à la fois le mappage des éléments XML à la structure d'analyse commune et la recherche XML native.

Tous les types et fonctions utilisés dans le fichier de configuration doivent être présentés dans la description du système de types de la procédure d'analyse personnalisée. Vous pouvez créer un descripteur de système de types dans votre environnement UIMA à l'aide du module d'extension Component Descriptor

Editor Eclipse. Ce dernier permet de créer un fichier de descripteur sans qu'il soit nécessaire de connaître la syntaxe XML requise.

Une fois que vous avez créé et testé l'analyse personnalisée, utilisez l'assistant de génération UIMA PEAR (Processing Engine ARchive) pour créer une archive contenant les fichiers d'analyse personnalisés incluant la description du système de types. Ensuite, vous pouvez télécharger l'archive d'analyse personnalisée et vos fichiers de mappage des éléments XML à la structure d'analyse commune dans la recherche d'entreprise à l'aide de la console d'administration pour la recherche d'entreprise.

Tâches associées

«Création d'un fichier de mappage des éléments XML à la structure d'analyse commune»

Création d'un fichier de mappage des éléments XML à la structure d'analyse commune

Dans un fichier de mappage des éléments XML à la structure d'analyse commune, vous pouvez utiliser la gamme complète des options de configuration pour le mappage des types de données XML à UIMA.

A propos de cette tâche

Le fichier de mappage des éléments XML à la structure d'analyse commune est présenté dans l'exemple suivant.

L'exemple de rapport de police comporte des balises XML pour le type de crime, la date du crime, le lieu du crime, l'officier de police ayant effectué le rapport, le commissariat dont dépend l'officier de police, la description du suspect et un résumé des faits. Tous ces éléments sont suivis d'une section body. Par exemple :

```
<report>
 <doc>
  <crimeType>Vol de voiture</crimeType>
  <crimeDate>23/04/05 21h23</crimeDate>
  <crimeLocation>27 Main Street, Brynston, Springfield, New Jersey</crimeLocation>
  <reportingOfficer rank="Lt">Jakob
   <lastName>Collins</lastName>
  </reportingOfficer>
  <policePrecinct>Commissariat du 14e</policePrecinct>
  <suspectDescription>Homme, brun, lunettes noires,
    pantalon jeans avec une veste foncée sans doute noire,
    </suspectDescription>
  <abstract>Une Mercedes CLK a été volée le 23/04/2005 sur
   un parking devant le restaurant Blue Lagoon,
   27 Main Street, Brynston.(serial number: 32 2761 50871)</abstract>
  <body>Une Mercedes CLK a été volée le 23/04/2004 sur
   un parking devant le restaurant Blue Lagoon, 27 Main Street,
   Brynston.(numéro de série : 32 2761 50871)

Elle est de couleur noir et a des pneus Michelin.

Des témoins devant le restaurant ont vu des individus de sexe masculin
portant des vêtements de couleur sombre s'enfuir au volant de la voiture.
La voiture a été retrouvée abandonnée à l'adresse suivante à Brooklyn : Aliway Ave.
Le réservoir d'essence était vide.
Les sièges ont été tachés et les sièges arrière saccagés.
Aucun objet n'a été volé dans la voiture....</body>
 </doc>
```

```
 <image>
  <--! image of the crime scene as a base64-encoded string -->
 </image>
</report>
```

Un fichier de mappage des éléments XML à la structure d'analyse commune créé à partir de cet exemple de rapport pourrait avoir la structure suivante. L'exemple utilise le système de types défini pour le scénario de rapport de police.

```
<?xml version="1.0"?>
<xmlCasInitializerConfiguration
 xmlns="http://www.ibm.com/2005/uima/jedii_ci_xml">

 <identifier>Valeur par défaut</identifier>
 <description>Exemple de configuration</description>

 <contentElements>
   <element>/report/doc</element>
 </contentElements>

 <elementToTypeMappings>
   <elementToTypeMapping>
     <element>//doc//reportingOfficer</element>
     <type>com.ibm.omnifind.types.Person</type>
     <featureValueAssignment>
       <feature>role</feature>
       <basicValue default="Reporting officer">
           </basicValue>
     </featureValueAssignment>
     <featureValueAssignment>
       <feature>gender</feature>
       <basicValue default="male"
          useAttributeValue="sex"/>
     </featureValueAssignment>
     <featureValueAssignment>
       <feature>surName</feature>
       <values concatenate="true" delimiter=" ">
         <basicValue useAttributeValue="rank"
                 default="Lt"/>
         <basicValue useElementContent="lastName"/>
       </values>
     </featureValueAssignment>
   </elementToTypeMapping>
   <elementToTypeMapping>
     <element>//doc</element>
     <type>com.ibm.omnifind.types.PoliceReport</type>
     <featureValueAssignment>
       <feature>crimeDescription</feature>
       <basicValue useElementContent="abstract"
                trim="true">
            </basicValue>
     </featureValueAssignment>
   </elementToTypeMapping>
 </elementToTypeMappings>

</xmlCasInitializerConfiguration>
```

Restrictions

Le fichier de mappage est fractionné en deux sections :

Elément <contentElements>

Utilisez cet élément si vous souhaitez une extraction de contenu spécifique. L'exemple de fichier de mappage extrait le contenu de la section <doc> d'un document et ignore les autres sections du document. Dans le rapport de police XML, l'image peut être de grande taille et peu utile pour le

traitement de texte. En indiquant <doc> en tant qu'élément de contenu et non <image>, l'image est supprimée avant qu'aucun traitement de texte ne commence.

<elementToTypeMappings>
Utilisez cet élément pour indiquer quels éléments XML individuels (indiqué dans un élément <elementToTypeMapping>) du document doivent être mappés à quelles structures de fonctions dans la structure d'analyse commune.

Si vous utilisez l'option d'extraction de contenu, les éléments XML indiqués dans la section <elementToTypeMappings> doivent être inclus dans les éléments XML indiqués dans la section <contentElements>.

Procédure

Pour créer un fichier de mappage des éléments XML à la structure d'analyse commune :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML pour valider les éléments XML. Le schéma XSD du fichier de mappage est appelé XMLCasInitSchema.xsd et se trouve dans votre installation de recherche d'entreprise à l'emplacement *RACINE_INSTALL_RECHERCHE_ENTREPRISE*/packages/uima/configuration_xsd/.
2. Incluez vos mappages dans un élément <xmlCasInitializerConfiguration xmlns="http://www.ibm.com/2005/uima/jedii_ci_xml">. L'espace de nom (indiqué dans l'attribut xmlns) doit être exactement identique à celui affiché.
3. Ajoutez un élément <contentElements> si vous souhaitez extraire un contenu spécifique des sections du document et un élément <elementToTypeMappings> qui définit quel élément XML individuel du document doit être mappé à quelles structures de fonctions dans la zone d'analyse commune.
4. Ajoutez un élément <identifier> et un élément <description>. L'identificateur détermine quel mappage doit être utilisé avec quel document XML. L'identificateur doit contenir l'élément principal du document, tel que doc. Si l'identificateur a la valeur par défaut, l'élément principal du document est inapproprié et le mappage est appliqué à tout document XML.
5. Ajoutez un élément <contentElements> si vous souhaitez extraire les informations qui se trouvent uniquement dans les parties appropriées d'un document. Il comporte l'élément de composant suivant :
   - Un ou plusieurs éléments <element> qui contiennent le chemin d'un élément XML du document et qui respectent la syntaxe XPath, par exemple <element>/doc/crimeType</element>.
6. Ajoutez un élément <elementToTypeMappings> si vous souhaitez indiquer quels éléments XML du document doivent être mappés à quelles structures de fonctions dans la structure d'analyse commune. Il comporte les éléments de composant suivants :
   - Un ou plusieurs éléments <elementToTypeMapping>. Cet élément doit avoir les éléments imbriqués suivants :
     – Un élément <element> permettant de définir le chemin d'un élément XML et respectant la syntaxe XPath. Une barre oblique en début de chaîne signifie qu'un chemin complet a été indiqué. Par exemple, abstract sous l'élément principal doc. Deux barres obliques (//) correspondent à un sous-ensemble de chemins. Par exemple, birthDate doit être placé dans reportingOfficer même si d'autres éléments peuvent être placés entre ces deux éléments.

- Un élément <type>, qui indique un type défini dans la description de système de types. Il doit être de type Annotation.
- Aucun élément <featureValueAssignment> ou plusieurs éléments de ce type.

7. Dans un élément <featureValueAssignment>, attribuez un nom à la fonction de type String dans l'élément <feature> et attribuez une valeur dans l'élément <basicValue>. Plusieurs éléments <basicValue> peuvent être ajoutés entre un élément <values>.

   L'élément <basicValue> peut avoir des attributs. Ces attributs peuvent être useAttributeValue, useElementContent, default et trim.

Utilisez useAttributeValue si vous voulez utiliser la valeur d'un attribut en tant que valeur d'une fonction. Exemple :

```
<elementToTypeMapping>
  <element>/doc//reportingOfficer</element>
  <type>com.ibm.omnifind.types.Person</type>
  <featureValueAssignment>
    <feature>role</feature>
    <basicValue default="Reporting officer"/>
  </featureValueAssignment>
  <featureValueAssignment>
    <feature>gender</feature>
    <basicValue default="male" useAttributeValue="sex"/>
  </featureValueAssignment>
</elementToTypeMapping>
```

produit la sortie suivante :

- Pour chaque balise XML <reportingOfficer> détectée dans une balise XML <doc> du document, une structure de fonctions de type com.ibm.omnifind.types.Person est créée.
- Si la balise <reportingOfficer> contient un attribut sex, la fonction gender de la structure de fonctions nouvellement créée a la valeur de l'attribut.

Utilisez l'attribut useElementContent pour ajouter le contenu en tant que valeur d'une fonction. Par exemple, dans le fragment de mappage suivant :

```
<elementToTypeMapping>
  <element>//doc</element>
  <type>com.ibm.omnifind.types.PoliceReport</type>
  <featureValueAssignment>
    <feature>crimeDescription</feature>
    <basicValue useElementContent="abstract" trim="true"/>
  </featureValueAssignment>
</elementToTypeMapping>
```

le texte couvert par l'élément <abstract> dans <doc> devient la valeur de la structure de fonctions crimeDescription. Tous les espaces de fin et de début sont supprimés.

Plusieurs valeurs peuvent être indiquées entre l'élément <values> dans les situations suivantes :

- La fonction à configurer est de type StringArray.
- Un grand nombre de chaînes sont concaténées en une chaîne à l'aide de l'attribut delimiter et donc mappent à une fonction de type String. Par exemple, le titre Mr. est une constante, le prénom est la valeur d'un attribut et le nom de famille est couvert par un élément XML :

```
<elementToTypeMapping>
  <element>//doc//reportingOfficer</element>
  <type>com.ibm.omnifind.types.Person</type>
  <featureValueAssignment>
    <feature>surName</feature>
    <values concatenate="true" delimiter=" ">
      <basicValue default="Mr."/>
      <basicValue useAttributeValue="rank"
         default="Lt."/>
      <basicValue useElementContent="lastName"/>
    </values>
  </featureValueAssignment>
</elementToTypeMapping>
```

Les valeurs de la fonction de chaîne sont extraites du fichier de mappage en l'état. Les valeurs conservent tous les espaces de début et de fin. Toutefois, les espaces sont supprimés des noms de type et de fonction. Par exemple, `<type><b>com.ibm.omnifind.types.Person</b></type>` devient `<type><b>com.ibm.omnifind.types.Person</b></type>`.

Définissez les conditions des attributs à l'aide de l'élément `<condition>`. Par exemple, la structure de fonctions de type `com.ibm.omnifind.types.Person` est créée uniquement si `<suspectDescription>` se trouve dans le document avec l'attribut `armed` ayant la valeur `yes` :

```
<elementToTypeMapping>
  <element>//suspectDescription</element>
  <type>com.ibm.omnifind.types.Person</type>
  <condition attribute="armed"  value="yes"/>
</elementToTypeMapping>
```

En fonction de l'exemple de rapport de police et du fichier de mappage défini, les structures de fonctions suivantes sont créées :

com.ibm.omnifind.types.PoliceReport

- covered text: ″Vol de voiture 23/04/05 09:23 27 Main Street, Brynston, Springfield, New Jersey Jakob Collins 14e circonscription Individu de sexe masculin, cheveux foncés, lunettes, pantalon jeans avec une veste foncée sans doute noire, une Mercedes modèle CLK a été ... Aucun objet n'a été volé dans la voiture.
- begin = 2
- end = 904
- knownSuspects = null
- crimeDescription = ″Une Mercedes modèle CLK a été volée le 23 avril 2005 sur un parking devant le restaurant Blue Lagoon restaurant, 27 Main Street, Brynston.(numéro de série : 32 2761 50871)″

com.ibm.omnifind.types.Person

- covered text = ″Jakob Collins″
- begin = 112
- end = 127
- role = ″Reporting officer″
- firstName = null
- surName = ″Lt Collins″
- gender = ″male″

Une fois que vous avez créé le fichier de mappage, vous devez le télécharger dans la recherche d'entreprise et sélectionner le fichier de mappage des éléments XML à

la structure d'analyse commune ainsi que d'autres options d'analyse personnalisées dans la console d'administration de la recherche d'entreprise.

Concepts associés

«Marquage XML dans l'analyse et la recherche», à la page 28

Référence associée

«Exemple de description de système de types», à la page 25

Résultats de l'analyse de texte

Tous les résultats de l'analyse de texte sont stockés dans la structure d'analyse commune.

Les annotateurs placent et lisent des données dans la structure d'analyse commune. Les consommateurs de la structure d'analyse commune (*consommateurs CAS*) lisent seulement à partir de la structure d'analyse commune. Les consommateurs CAS effectuent le traitement final sur les résultats d'analyse stockés dans la structure d'analyse commune. La recherche d'entreprise contient deux clients CAS :

- Client qui indexe le contenu de la structure d'analyse commune dans un moteur de recherche. Ce consommateur requiert un fichier de mappage de la structure d'analyse commune à l'index que vous sélectionnez à l'aide de l'analyse de texte personnalisée sur la console d'administration de la recherche d'entreprise.
- Client qui charge des résultats d'analyse spécifiques dans une base de données relationnelle. Ce consommateur requiert également un fichier de mappage de la structure d'analyse commune à la base de données que vous sélectionnez à l'aide des options d'analyse de texte personnalisée sur la console d'administration de la recherche d'entreprise.

Si nécessaire, vous pouvez déployer des clients CAS personnalisés dans la recherche d'entreprise. Pour savoir comment créer un client, consultez la documentation UIMA. Pour savoir comment télécharger et utiliser votre consommateur dans la recherche d'entreprise, consultez le site Web IBM UIMA developerWorks à l'adresse http://www.ibm.com/developerworks/db2/zones/db2ii/.

Concepts associés

«Mappage d'index pour les résultats de l'analyse personnalisée», à la page 40

«Mappage de base de données pour les résultats de l'analyse sélectionnés», à la page 47

Chemins de fonctions

Un chemin de fonctions permet d'accéder à des valeurs de fonctions dans les structures d'analyse commune, de la même manière que les instructions XPath permettent d'accéder aux éléments XML d'un document XML.

Les chemins de fonctions sont utiles si vous souhaitez accéder à une structure de fonctions qui associe des fonctions complexes, par exemple des fonctions de valeurs de tableau ou qui désignent une autre structure de fonctions. A l'aide d'un chemin de fonctions, vous pouvez associer la valeur d'une fonction directement à une structure de fonctions et stocker cette valeur dans l'index de recherche sémantique ou dans une base de données.

Prenons l'exemple d'un annotateur qui identifie les voitures et leurs marques. Il crée des annotations de type `car` ayant un attribut `make`. Toutefois, l'attribut `make`

ne contient pas le nom réel de l'entreprise (par exemple, Chevrolet) mais une structure de fonctions de type `Company` qui contient elle-même un attribut `companyname`. Pour activer une requête sémantique qui associe des noms de voiture à des noms d'entreprise, un chemin de fonctions `make/companyname` permet d'associer la valeur de `companyname` à l'étendue car générée pour l'annotation car. Permet d'activer la requête ″Extraire les documents qui contiennent les voitures fabriquées par Chevrolet″ en utilisant '/car[@make=″Chevrolet″]'.

Un chemin de fonctions est une suite de noms de fonction (f1/.../fn) avec les propriétés suivantes :

- La valeur d'un chemin de fonctions peut être String, Integer, Float ou un tableau d'un de ces types.
- Toutes les fonctions de ce chemin de f1 à fn-1 doivent avoir un type complexe, c'est-à-dire de type `uima.cas.TOP`, `uima.cas.FSArray`, `uima.cas.FSList` ou d'un de ses sous-types.
- La dernière fonction du chemin peut inclure un type complexe. Elle peut également inclure un type ou un sous-type de `uima.cas.Float`, `uima.cas.Integer`, `uima.cas.String`, `uima.cas.FloatArray`, `uima.cas.IntegerArray`, `uima.cas.StringArray`, `uima.cas.FloatList`, `uima.cas.IntegerList` ou `uima.cas.StringList`.
- Vous pouvez également saisir une fonction. Le nom de la fonction doit être ajouté avant le nom du type complet et être séparé par le caractère deux points. Par exemple, `f1/com.ibm.es.SomeType:f2/.../fn`.

Vous pouvez restreindre la portée type d'une fonction particulière. Par exemple, utilisons une fonction `additionalInfo` de type `uima.cas.TOP`. Si vous savez que la valeur de la fonction `additionalInfo` est de type `EmployeeInfo` qui a la fonction `salary`, vous pouvez accéder à cette fonction en utilisant `additionalInfo/EmployeeInfo:salary`. Dans cet exemple, le chemin de fonctions `additionalInfo/salary` génère une erreur car `salary` n'a pas été défini pour le type `uima.cas.TOP`.

Les fonctions qui ont des valeurs de type tableau ou liste ont les propriétés supplémentaires suivants :

- Utilisez des crochets ([<number>]) pour sélectionner un certain élément dans le tableau ou dans la liste. Un tableau commence à zéro (0). Par exemple, pour sélectionner le premier élément du tableau des entreprises, utilisez companies[0]. Le marqueur spécial [last] peut permettre de sélectionner la dernière entrée d'un tableau, quelle que soit sa taille, par exemple companies[last].
- Utilisez des crochets vides ([]) pour obtenir tous les éléments. Un seul crochet vide ([]) est admis dans un chemin de fonctions. Par exemple, dans un tableau de suspects, le chemin de fonctions `knownSuspects[]/com.ibm.omnifind.types.Suspect:surName` rassemble tous les noms des suspects dans un tableau String.
- Lorsqu'un chemin de fonctions qui renvoie un tableau est utilisé lors de l'indexation, les éléments de tableau sont concaténés (séparés par des espaces) et placés dans l'index en tant qu'attribut unique ou comportant plusieurs termes ou que zone.
- Vous devez entrer l'élément suivant du chemin de fonctions. Le nom de type est le type d'éléments du tableau. Par exemple, utilisons une structure de fonctions de type `Info`. Ce type a une fonction nommée `companies`, dont la plage est un élément FSArray. Les éléments du tableau sont de type `Company`. `Company` a une fonction nommée `profit`. Pour obtenir le bénéfice de la troisième entreprise, entrez (à l'aide des noms de type complets) `companies[2]/Company:profit`.

Fonctions intégrées

Les fonctions intégrées sont des noms de fonction prédéfinies avec des sémantiques particulières. Elles peuvent permettre d'accéder aux informations qui ne sont pas contenues dans la structure de fonctions elles-mêmes, par exemple, le type de la structure de fonctions ou le texte couvert d'une annotation. Vous pouvez les utiliser dans un chemin de fonctions en tant que dernier ou seul élément.

Les fonctions intégrées suivantes peuvent être utilisées dans les deux fichiers de configuration de mappage :

- `fsId()` renvoie l'ID de la structure de fonctions. L'ID renvoyé est un entier (32 bits). Utilisez cette fonction intégrée pour accéder aux parties d'un document qui correspondent exactement à la requête.
- `typeName()` renvoie le type d'objet de structure d'analyse commune sous la forme de chaîne. Le type correspond au nom de type complet incluant des préfixes d'espace de nom, par exemple `uima.tcas.Annotation`. Dans un contexte de base de données, `typeName()` est particulièrement utile lorsque vous stockez des types et des sous-types dans la même colonne et que vous souhaitez connaître un type réel d'une annotation ou d'une structure de fonctions. L'exemple suivant stocke le type de personne, tel que *suspect* ou *witness*, dans la colonne des rôles.

```
<explicitMappingRule  applyToSubTypes="false">
   <type>com.ibm.omnifind.types.Person</type>
   <table>sample.person</table>
   <featureMappings>
     <featureMapping>
       <feature>typeName()</feature>
       <column>role</column>
     </featureMapping>
   </featureMappings>
 </explicitMappingRule>
```

- `coveredText()` renvoie le texte couvert par l'objet d'analyse commune. `coveredText()` est disponible uniquement pour les annotations et leurs sous-types. N'utilisez pas cette fonction intégrée sur les structures de fonction qui ne sont pas incluses dans une classification par le type d'annotation. L'exemple suivant stocke le nom d'un suspect dans la colonne suspectName.

```
<implicitMappingRule applyToSubTypes="false">
   <type>com.ibm.omnifind.types.Suspect</type>
   <relation>sample.person</relation>
   <featureMappings>
     <featureMapping>
       <feature>coveredText()</feature>
       <column>suspectName</column>
       <length>128</length>
     </featureMapping>
   </featureMappings>
 </implicitMappingRule>
```

- `[]` renvoie un descripteur de l'entrée de conteneur en cours (tableau ou liste). La fonction implique une itération, ce qui signifie qu'une entrée est créée dans l'index ou la table de base de données pour chaque élément dans le tableau ou la liste. L'exemple suivant est extrait d'un fichier de mappage de la structure d'analyse commune à la base de données dans lequel la fonction intégrée [:index] est admise.

```
<implicitMappingRule applyToSubTypes="false">
   <type>uima.cas.FSArray</type>
   <table>sample.knownSuspects</table>
   <featureMappings>
     <featureMapping>
       <feature>uniqueId()</feature>
```

```
      <column>arrayId</column>
    </featureMapping>
    <featureMapping>
      <feature>[:index]</feature>
      <column>arrayIndex</column>
    </featureMapping>
    <featureMapping>
      <feature>[]/com.ibm.omnifind.types.Suspect:uniqueId()</feature>
      <column>suspectId</column>
    </featureMapping>
  </featureMappings>
</implicitMappingRule>
```

Les fonctions intégrées suivantes peuvent être utilisées uniquement dans le fichier de mappage de la structure d'analyse commune à la base de données :

- `uniqueId()` renvoie l'ID unique global de la structure de fonctions. L'ID unique renvoyé est une chaîne de longueur fixe (27 caractères) et une concaténation du résultat de `fsId()`, `docId()`, `docTimestamp()` et du nombre de segments car les documents peuvent être morcelés en plusieurs structures d'analyse commune dans la recherche d'entreprise.

  La chaîne renvoyée peut inclure des caractères compris dans les plages ″a-z″ et ″A-Z″, les nombres ″0-9″, le point-virgule (″;″) et le caractère deux points (″:″).

  Le résultat de l'élément `uniqueId()` peut être utilisé en tant que clé principale pour les tables.
- `objectId()` renvoie l'ID de l'annotation ou de la structure de fonctions. `objectId()` est similaire à `uniqueId()` mais il ne contient pas le résultat de `docTimestamp()`. L'ID renvoyé est unique uniquement dans une collection dans laquelle les documents sont analysés une seule fois. Si tous les documents et versions de document doivent être uniques, vous devez utiliser `uniqueId()`.

  La chaîne renvoyée de la fonction intégrée `objectId()` a une longueur fixe de 16 caractères et peut inclure des caractères compris dans les plages ″a-z″ et ″A-Z″, les nombres ″0-9″, le point-virgule (″;″) et le caractère deux points (″:″).

  Si `uniqueId()` ou `objectId()` référence des structures de fonctions vides, la valeur par défaut définie dans la définition de table de base de données est utilisée, aucun objet vide d'un type référencé n'est stocké.
- `docId()` renvoie l'ID document. La valeur renvoyée est de type integer (32 bits).

  L'exemple suivant affiche ces fonctions intégrées :

```
<explicitMappingRule applyToSubTypes="true">
   <type>com.ibm.omnifind.types.PoliceReport</type>
   <table>sample.PoliceReport</table>
   <featureMappings>
     <featureMapping>
       <feature>uniqueId()</feature>
       <column>policeReportId</column>
     </featureMapping>
     <featureMapping>
       <feature>docId()</feature>
       <column>policeReportDocId</column>
     </featureMapping>
   </featureMappings>
 </explicitMappingRule>
```

- `docUri()` renvoie l'URI du document.
- `docTimestamp()` renvoie l'heure (en millisecondes) à laquelle le document a été traité. Cette fonction intégrée est utile pour le suivi des versions de document, par exemple, si vous souhaitez savoir si la version de document que vous utiliser est la dernière à avoir été transmise par le moteur de balayage.

```
<explicitMappingRule applyToSubTypes="false">
   <type>com.ibm.omnifind.types.PoliceReport</type>
   <relation>sample.PoliceReport</relation>
   <featureMappings>
     <featureMapping>
       <feature>uniqueId()</feature>
       <column>policeReportId</column>
     </featureMapping>
     <featureMapping>
       <feature>docTimestamp()</feature>
       <column>reportVersion</column>
     </featureMapping>
   </featureMappings>
 </explicitMappingRule>
```

- `parentId()` renvoie l'élément `fsId()` de la structure de fonctions qui comporte un mappage de conteneur. `parentId()` est valide uniquement dans le contexte d'un mappage de conteneur.
- `uniqueParentId()` renvoie l'élément `uniqueId()` de l'annotation ou de la structure de fonctions comprise dans un mappage de conteneur. La fonction intégrée est valide uniquement dans le contexte d'un mappage de conteneur.
- `[:index]` renvoie l'index de l'entrée de conteneur en cours (tableau ou liste).

Tâches associées

«Extraction des parties d'un document qui correspondent à une requête de recherche sémantique», à la page 58

Filtres

Les filtres permettent de restreindre les règles de mappage dans les fichiers de mappage de la structure d'analyse commune à l'index et fichiers de mappage de la structure d'analyse commune à la base de données. Les résultats de l'analyse sont ajoutés à l'index ou à une table JDBC uniquement si le filtre est appliqué.

L'élément `<filter>` est facultatif et il permet de restreindre les mappages aux fonctions qui ont une certaine valeur d'attribut. Cet élément est utile si vous voulez qu'un attribut se comporte comme commutateur pour les éléments à indexer ou à ajouter à la base de données. Par exemple, les personnes et les entreprises peuvent être enregistrées dans une annotation de type `EntityAnnotation`. Sa fonction appelée `type` a la valeur `person` ou `organization`. Pour extraire uniquement les personnes et non les entreprises, vous pouvez ajouter le filtre suivant à la règle de mappage :

```
<filter syntax="FeatureValue">type = "person"</filter>
```

Chaque expression de filtre a la forme :

```
<FeaturePath> <Operator> <Literal>
```

où :

- FeaturePath est un chemin de fonctions dans la structure d'analyse commune.
- Operator est =, !=, <, <=, > ou >=. < (et uniquement <) doit être exprimé sous la forme de `&lt;`.
- Le littéral est un entier, un nombre à virgule flottante (aucun syntaxe d'exposant n'est prise en charge) ou un un littéral de chaîne placé entre guillemets.

`<FeaturePath>`, `<Operator>` et `<Literal>` doivent être séparés par un espace.

Les exemples suivants sont des filtres valides :

- `<filter syntax="FeatureValue"> foo = "hello world" </filter>`

La fonction foo contient la chaîne hello world.

- `<filter syntax="FeatureValue"> foo &lt; 42 </filter>`

  La fonction foo présente une valeur d'entier inférieure à 42.
- `<filter syntax="FeatureValue"> make/company = "Chevrolet" </filter>`

  Le chemin de fonctions make/company dans lequel la fonction make contient une structure de fonctions qui a une fonction company avec la valeur Chevrolet.
- `<filter syntax="FeatureValue"> bar7 >= 0.5 </filter>`

  La fonction bar7 présente une valeur flottante supérieure ou égale à 0,5.

Mappage d'index pour les résultats de l'analyse personnalisée

Après avoir exécuté l'analyse personnalisée sur une collection de documents, vous pouvez utiliser le moteur de recherche dans la recherche d'entreprise afin de générer un index à partir des informations stockées dans la structure d'analyse commune créée par les algorithmes d'analyse personnalisée.

Le mappage des résultats de l'analyse à des zones, des étendues de texte et des attributs dans l'index de recherche d'entreprise permet d'utiliser ces informations dans les requêtes. L'association de l'analyse personnalisée à la recherche d'entreprise pouvant indexer à la fois des mots et des étendues de texte active la recherche d'entreprise.

A l'aide du fichier de mappage de la structure d'analyse commune à l'index, vous pouvez déterminer les résultats de l'analyse de la structure d'analyse commune à indexer.

Vous pouvez utiliser différents styles pour mapper des structures de fonction de la structure d'analyse commune à l'index de recherche d'analyse.

Annotation

Si vous indexez des structures de fonction dans la structure d'analyse commune à l'aide du style d'annotation, toutes les annotations des types spécifiés sont stockées dans l'index en tant qu'étendues pouvant être recherchées.

Par exemple, si une structure de fonctions qui étend une certaine zone de texte est de type `person` et qu'elle est indexée à l'aide du style d'annotation, les requêtes suivantes sont possibles :

Tableau 2. Requêtes exemple

Informations requises	Requête possible
Extraire tous les documents qui contiennent au moins un nom de personne	<person/>
Extraire tous les documents dans lequel un supérieur hiérarchique est indiqué dans une annotation de personne	<person>boss</person>
Extraire tous les documents dans lesquels la mention de langue est indiquée dans la même phrase qu'un de mes concurrents	<sentence><person>Lang</person> <competitor/></sentence>

Les attributs des structures de fonctions sont également indexés comme partie de l'étendue. Prenons l'exemple d'un annotateur qui détecte les

voitures et stocke la marque de voiture en tant que fonction `make` de l'annotation `car`. Permet d'activer le type suivant de requête : ″Extraire les documents qui mentionnent les voitures de marque Chevrolet″.

Zone Utilisez ce style si vous souhaitez que le contenu des structures de fonctions soient accessibles lors de la recherche en utilisant les fonctions de recherche de zone de la recherche d'entreprise. De cette manière, le contenu d'une structure de fonctions peut être affiché dans les résultats de la recherche ou utilisé dans la recherche paramétrique.

Par exemple, si vous mappez des dosages de médicaments à une zone paramétrique, vous pouvez utiliser la requête suivante : ″Extraire tous les documents mentionnant un médicament pris à un dosage supérieur à 100 milligrammes″.

Interruption
Utilisez ce style si vous souhaitez qu'une structure de fonctions particulière soit interprétée en tant que délimiteur, par exemple des sections ou des paragraphes. La recherche d'entreprise détecte les phrases et les paragraphes par défaut. Utilisez ce style uniquement si l'analyse personnalisée détecte des éléments structurels supplémentaires dans un document que vous souhaitez interpréter différemment.

Vous pouvez également avoir recours aux résultats de l'analyse pour influencer le classement des documents dans la recherche d'entreprise, même pour des requêtes de mot clé simples. Cette action s'effectue en deux étapes :

1. Mappage des structures de fonction à des étendues ou à des zones pouvant être recherchées, à l'aide du style de mappage d'annotation ou de zone.
2. Définition d'une classe à l'aide de la console d'administration de recherche d'entreprise et mappage du nom de zone ou d'étendue à cette classe de pondération.

Si l'utilisateur entre un terme de recherche contenu dans la structure de fonctions, le classement du document est plus élevé. Prenons l'exemple d'un annotateur qui identifie les personnes et les noms d'entreprise. En mappant ces structures de fonctions à des étendues (″person″ et ″company″, par exemple) puis en mappant ces étendues aux classes de pondération, le résultat de recherche ″gap″ classe les documents parlant de l'entreprise ″Gap″ à un niveau plus élevé que ceux contenant à peine le terme ″gap″.

Une fois que vous avez créé le fichier de mappage de la structure d'analyse commune à l'index, vous pouvez le télécharger dans la recherche d'entreprise à l'aide de la console d'administration.

Tâches associées

«Création du fichier de mappage de la structure d'analyse commune à l'index»

Création du fichier de mappage de la structure d'analyse commune à l'index

A l'aide du fichier de mappage de la structure d'analyse commune à l'index, vous pouvez déterminer les résultats de l'analyse de la structure d'analyse commune que vous voulez indexer pour activer la recherche.

A propos de cette tâche

Le fichier de mappage de la structure d'analyse commune à l'index est en XML. L'exemple de fichier de mappage de la structure d'analyse commune à l'index se fonde sur le système de types défini pour le scénario de rapport de police.

```
<?xml version="1.0" encoding="UTF-8"?>
<indexBuildSpecification
xmlns="http://www.ibm.com/of/822/consumer/index/xml">
 <skipCondition>
   <type>com.ibm.uima.tt.DocumentAnnotation</type>
   <filter syntax="FeatureValue">toBeprocessed = 0</filter>
 </skipCondition>

 <indexBuildItem>
   <name>com.ibm.omnifind.types.Person</name>
   <indexRule>
     <style name="Annotation">
       <attributemappings>
         <mapping>
           <feature>role</feature>
           <indexName>role</indexName>
         </mapping>
         <mapping>
           <feature>title</feature>
           <indexName>title</indexName>
         </mapping>
         <mapping>
           <feature>gender</feature>
           <indexName>gender</indexName>
         </mapping>
       </attributemappings>
     </style>
   </indexRule>
 </indexBuildItem>
 <indexBuildItem>
   <name>com.ibm.omnifind.types.Suspect</name>
   <indexRule>
     <style name="Annotation"/>
     <style name="Field">
       <attribute name="parametric" value="false"/>
       <attribute name="fieldSearchable"
         value="true"/>
       <attribute name="returnable" value="true"/>
     </style>
   </indexRule>
 </indexBuildItem>
 <indexBuildItem>
   <name>com.ibm.omnifind.types.City</name>
   <indexRule>
       <style name="Annotation">
       <attributemappings>
         <mapping>
           <feature>cityDistrict</feature>
           <indexName>district</indexName>
         </mapping>
       </attributemappings>
     </style>
   </indexRule>
 </indexBuildItem>
 <indexBuildItem>
   <name>com.ibm.omnifind.types.Date</name>
   <indexRule>
     <style name="Field">
       <attribute name="fixedName" value="Date"/>
       <attribute name="fieldSearchable"
           value="true"/>
       <attribute name="returnable" value="true"/>
     </style>
```

```
    <style name="Field">
      <attribute name="fixedName" value="hour"/>
      <attribute name="valueFeature"  value="hour"/>
      <attribute name="parametric" value="true"/>
    </style>
  </indexRule>
  <filter syntax="FeatureValue">year="2005"</filter>
</indexBuildItem>
<indexBuildItem>
  <name>com.ibm.omnifind.types.PoliceReport</name>
  <indexRule>
    <style name="Annotation">
      <attribute name="fixedName"
         value="PoliceReport"/>
      <attributemappings>
        <mapping>
          <feature>crimeDescription</feature>
          <indexName>crimeDescription</indexName>
        </mapping>
        <mapping>
          <feature>time/coveredText()</feature>
          <indexName>time</indexName>
        </mapping>
        <mapping>
          <feature>date/englDate</feature>
          <indexName>date</indexName>
        </mapping>
        <mapping>
          <feature>location/coveredText()</feature>
          <indexName>location</indexName>
        </mapping>
        <mapping>
          <feature>knownSuspects[]/com.ibm.omnifind.types.Suspect:surName</feature>
          <indexName>suspectsLastNames</indexName>
        </mapping>
      </attributemappings>
    </style>
  </indexRule>
</indexBuildItem>
</indexBuildSpecification>
```

Restrictions

Le fichier de mappage de la structure d'analyse commune à l'index doit contenir l'ensemble des résultats de l'analyse que vous souhaitez pouvoir rechercher dans les requêtes.

Procédure

Pour créer le fichier de mappage de la structure d'analyse commune à l'index :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix. Le schéma XSD du fichier de mappage est appelé `CasToIndexMapping.xsd` et se trouve dans votre installation de recherche d'entreprise à l'emplacement *RACINE_INSTALL_RECHERCHE_ENTREPRISE*`/packages/uima/configuration_xsd/`.
2. Incluez vos mappages dans un élément `<indexBuildSpecification xmlns="http://www.ibm.com/of/822/consumer/index/xml">`. L'espace de nom (indiqué dans l'attribut xmlns) doit être exactement identique à celui affiché.
3. Ajoutez un élément `<skipCondition>` pour empêcher que certains documents soient indexés, en fonction d'une certaine valeur de fonction. Cet élément est facultatif. Dans l'exemple, les documents contenant une structure de données

de type com.ibm.uima.tt.DocumentAnnotation avec une fonction nommée toBeProcessed ayant la valeur zéro ne seront pas indexés.

4. Ajoutez un ou plusieurs éléments <indexBuildItem> qui contiennent le mappage d'une structure de fonctions particulière dans la structure d'analyse commune à une structure de l'index.
5. Sauvegardez et validez le fichier XML.

Elément <indexBuildItem>

Le fichier de mappage de la structure d'analyse commune à l'index contient un ou plusieurs éléments <indexBuildItem>. Chaque élément décrit le mappage d'une structure de fonctions particulière dans la structure d'analyse commune à une structure de l'index (une étendue ou une zone).

L'élément <name> contient le type de structure de fonctions. Il existe deux méthodes permettant de définir un type :

- Nom de type complet. Par exemple, com.ibm.omnifind.types.Suspect
- Caractère générique. Par exemple, com.ibm.omnifind.types.*. Le caractère générique peut être ajouté uniquement à la fin de la spécification de type.

Utilisez uniquement des sous-types uima.tcas.Annotation en tant qu'éléments de génération d'index. Si une fonction est un sous-type uima.cas.TOP (et non uima.tcas.Annotation), vous pouvez accéder à cette structure de fonctions à l'aide d'un chemin de fonctions commençant à partir d'une annotation.

Si le type A est un sous-type du type B (dans l'exemple, com.ibm.omnifind.types.Suspect en tant que sous-type de com.ibm.omnifind.types.Person) et qu'il existe des éléments <indexBuildItem> Ia et Ib définis pour les deux types, le traitement s'effectue comme suit :

- Chaque règle d'index définie dans Ib est appliquée aux structures de fonctions de type B et aux structures de traits de type A
- Chaque règle d'index définie dans Ia est appliquée aux structures de fonctions de type A

Dans l'exemple, l'élément <indexBuildItem> défini pour les annotations com.ibm.omnifind.types.Person s'applique également aux annotations com.ibm.omnifind.types.Suspect. Deux étendues sont créées pour une annotation suspect : Person et Suspect.

L'élément <filter> est facultatif et il permet de restreindre le mappage <indexBuildItem> aux structures de fonctions qui ont une certaine valeur d'attribut. Cet élément est utile si vous voulez qu'un attribut se comporte comme commutateur pour les éléments à indexer. Par exemple, les personnes et les entreprises peuvent être enregistrées dans une annotation de type EntityAnnotation. Sa fonction appelée type a la valeur person ou organization. Pour extraire uniquement les personnes et non les entreprises, vous pouvez ajouter le filtre suivant :

```
<filter syntax="FeatureValue">type = "person"</filter>
```

Vous pouvez également choisir d'indexer les personnes et les entreprises en utilisant des noms d'étendue différents, par exemple person et organization. Pour cela, définissez deux éléments <indexBuildItem> de type EntityAnnotation et utilisez deux filtres sur la fonction type pour déclencher les personnes ou les entreprises.

Elément <indexRule>

Chaque élément <indexBuildItem> contient un élément <indexRule>. Chaque élément <indexRule> contient l'ensemble des informations requises pour le mappage d'une structure de fonctions se trouvant dans la structure d'analyse commune à l'index en tant que zone, annotation et style d'interruption. Les styles de zone et d'annotation prennent en charge plusieurs attributs. Vous ne pouvez pas utiliser le terme style, qui est pris en charge dans le SDK UIMA pour la recherche d'entreprise (le terme style est ignoré).

Pour les styles d'annotation et de zone, il existe les alternatives suivantes lorsque vous indiquez le nom d'annotation ou de zone dans l'index :

- Utilisez fixedName si vous souhaitez que chaque structure de fonctions soit accessible dans l'index portant le même nom. Dans l'exemple suivant, chaque structure de fonctions de type com.ibm.omnifind.types.Person sera mappée à une étendue nommée ″Person″ dans l'index.

  ```
  <indexBuildItem>
     <name>com.ibm.omnifind.types.Person</name>
     <indexRule>
       <style name="Annotation">
         <attribute name="fixedName" value="Person" />
       </style>
     </indexRule>
  </indexBuildItem>
  ```

  Active des requêtes telles ″Extraire des documents dans lequel un supérieur hiérarchique est inclus en tant que nom de personne″. La requête est exprimée de la manière suivante à l'aide de fragments XML : @xmlf2::'<Person>Boss</Person>'.

- Utilisez nameFeature si l'annotation stocke des entités différentes auxquelles vous souhaitez pouvoir accéder à l'aide de différentes étendues en fonction de la valeur d'une certaine fonction de l'annotation. Dans l'exemple suivant, com.ibm.tt.EntityAnnotation est indexé en tant qu'étendue person ou organization, selon la valeur de la fonction nommée type. La fonction peut également être un chemin de fonctions.

  ```
  <indexBuildItem>
     <name>com.ibm.tt.EntityAnotation</name>
     <indexRule>
       <style name="Annotation">
         <attribute name="nameFeature" value="type" />
       </style>
     </indexRule>
  </indexBuildItem>
  ```

  Active des requêtes, telles ″Extraire des documents sur l'entreprise WHO″ (en opposition au terme anglais ″who″). La requête est exprimée de la manière suivante dans la syntaxe XPath limitée : @xmlp::'/organization[ftcontains="WHO"]'.

- Si aucun des attributs ci-dessus n'est utilisé, le nom abrégé du type d'annotation de l'élément <indexBuildItem> est utilisé. Il s'agit de la valeur par défaut. Par exemple :

  ```
  <indexBuildItem>
     <name>com.ibm.uima.tutorial.RoomNumber</name>
     <indexRule>
       <style name="Annotation" />
       <style name="Field" />
     </indexRule>
   </indexBuildItem>
  ```

Cet élément <indexBuildItem> génère des annotations et des zones appelées RoomNumber chargées avec le texte couvert par com.ibm.uima.tutorial.RoomNumber.

Elément <style name="Annotation" />

L'annotation de l'élément <style> indique comment accéder aux informations d'étendue dans la recherche d'entreprise. Outre l'utilisation des attributs fixedName et nameFeature, ce style prend également en charge l'élément <attributemappings>. Dans cet élément, il est possible de mapper la valeur d'une fonction à un attribut de l'étendue en résultant dans l'index, que vous pouvez ensuite utiliser dans une expression de recherche.

Chaque mappage est effectué dans un élément <mapping> séparé. L'élément <feature> contient un chemin de fonctions et l'élément <indexName> contient le nom de l'attribut utilisé dans l'index pour stocker la valeur de <feature>. Par exemple,

```
<mapping>
  <feature>make/companyname</feature>
  <indexName>company</indexName>
</mapping>
```

Cet élément <mapping> stocke la valeur de la fonction du chemin make/companyname directement dans l'attribut d'index company.

Le mappage des valeurs de fonction aux attributs d'index est particulièrement utile si le système de types utilisé lors de l'analyse de texte est complexe, y compris un grand nombre de structures de fonctions imbriquées. A l'aide de l'élément <mapping>, des attributs appropriés peuvent être exposés, ce qui vous permet de les utiliser dans des requêtes sans bien connaître la structure du système de types d'origine.

Elément <style name="Field" />

La zone de l'élément <style> indique comment accéder aux informations de zone dans la recherche d'entreprise. Outre les attributs fixedName et nameFeature, vous pouvez définir les attributs suivants.

parametric
Si la valeur de la zone est true, elle peut être recherchée de manière paramétrique, par exemple, #dosage:>100

fieldSearchable
Si la valeur de la zone est true, elle peut être utilisée dans la recherche, par exemple, make:Bayer

returnable
Si la valeur de la zone est true, la zone et ses valeurs sont renvoyées dans le résultat de la recherche

Il est toujours possible de rechercher le contenu des informations de zone. Autrement dit, les informations de zone sont accessibles dans les recherches de mot clé normales.

L'attribut facultatif valueFeature définit la valeur de fonction à utiliser en tant que valeur de zone. Si la structure de fonctions est une annotation et que l'attribut n'est pas défini, le texte couvert de l'annotation est utilisé en tant que valeur de zone. Dans l'exemple,

```
<indexBuildItem>
   <name>com.ibm.omnifind.types.Date</name>
   <indexRule>
     <style name="Field">
       <attribute name="fixedName" value="date"/>
       <attribute name="fieldSearchable"
           value="true"/>
       <attribute name="returnable" value="true"/>
     </style>
     <style name="Field">
       <attribute name="fixedName" value="hour"/>
       <attribute name="valueFeature"  value="hour"/>
       <attribute name="parametric" value="true"/>
     </style>
   </indexRule>
   <filter syntax="FeatureValue">year="2005"</filter>
</indexBuildItem>
```

deux zones sont générées pour `com.ibm.omnifind.types.Date`. Une zone nommée `date` contient le texte couvert, par exemple, 17:15. Une autre zone contient la valeur de l'attribut `hour`. Vous pouvez effectuer une requête ici de la manière suivante : 'hour::<17'.

Elément `<style name="Breaking" />`

La valeur Breaking de l'élément `<style>` n'inclut pas d'autres éléments.

Une fois que vous avez créé le fichier XML, vous devez le télécharger dans la recherche d'entreprise et sélectionner le fichier de mappage de la structure d'analyse commune à l'index ainsi que d'autres options d'analyse personnalisées à l'aide de la console d'administration de la recherche d'entreprise.

Concepts associés

«Mappage d'index pour les résultats de l'analyse personnalisée», à la page 40

«Chemins de fonctions», à la page 35

Référence associée

«Filtres», à la page 39

«Exemple de description de système de types», à la page 25

Mappage de base de données pour les résultats de l'analyse sélectionnés

Une fois que vous avez analysé vos documents dans la recherche d'entreprise, vous pouvez stocker les résultats de l'analyse de texte sélectionnés dans une base de données compatible JDBC.

Cette version prend en charge DB2 Universal Database version 8.2.2 (com.ibm.db2.jcc.DB2Driver Version 2.3) ou supérieure et Oracle 10g (oracle.jdbc.driver.OracleDriver Version 1.0).

Pour DB2 Universal Database et Oracle, vous pouvez choisir d'insérer les résultats de l'analyse directement dans la base de données ou de générer les fichiers de chargement propres à la base de données équivalents et le script correspondant qui exécute les commandes de chargement.

Le mappage des résultats de l'analyse à des tables d'une base de données vous permet d'utiliser ces informations dans les étapes de traitement de veille

économique ou d'accéder directement aux parties appropriées d'un document qui correspondent à une requête de recherche sémantique.

Le fichier de mappage de la structure d'analyse commune à la base de données contient des informations de configuration de connexion aux bases de données et décrit quels résultats de l'analyse personnalisée doivent être stockés dans quelles tables et colonnes. Les noms des tables et des colonnes du fichier de mappage doivent correspondre aux tables et aux colonnes créées dans la base de données.

Une fois que vous avez écrit le fichier de mappage de la structure d'analyse commune à la base de données, vous pouvez le télécharger dans la recherche d'entreprise à l'aide de la console d'administration.

Tâches associées

«Création du fichier de mappage de la structure d'analyse commune à la base de données», à la page 49

Stockage des résultats de l'analyse dans une base de données

Pour stocker les résultats de l'analyse sélectionnés dans une base de données compatible JDBC, vous devez créer le fichier de mappage de la structure d'analyse commune à la base de données qui définit les résultats d'analyse à stocker dans une base de données et les bibliothèques du pilote JDBC nécessaires doivent se trouver dans le chemin défini dans le fichier de mappage.

Pour stocker les résultats de l'analyse dans une base de données compatible JDBC, procédez comme suit :

1. Déterminez les résultats de l'analyse à stocker dans la base de données. Créez une base de données qui contient les tables avec toutes les colonnes nécessaires des types de données appropriés.
2. Dans un éditeur XML, créez le fichier de mappage de la structure d'analyse commune à la base de données avec les données de configuration de la base de données et les résultats de l'analyse à stocker. Pour déterminer les résultats de l'analyse à inclure dans le fichier de mappage, vous devez connaître le système de types sous-jacent utilisé lorsque les documents sont traités.
3. Placez les bibliothèques du pilote JDBC dans un répertoire du noeud de l'index dans lequel elles seront accessibles au système de recherche d'entreprise.
4. Téléchargez et sélectionnez le fichier de mappage en utilisant la console d'administration de recherche d'entreprise.

Utilisation des ensembles de fichiers de chargement

Vous pouvez soit stocker les résultats d'analyse directement dans une base de données compatible JDBC, soit configurer le traitement pour utiliser les ensembles de fichiers de chargement et charger les données dans une base de données à un moment ultérieur.

L'utilisation des ensembles de fichiers de chargement présente les avantages suivants :

- Au total, un ensemble de fichiers de chargement ne peut jamais dépasser la taille de fichier maximale prise en charge par le système d'exploitation.
- Vous pouvez démarrer le chargement des données dans une base de données dès qu'un ensemble de fichiers de chargement est plein et vous n'avez pas à arrêter et à redémarrer l'analyseur syntaxique de documents pour éviter les conflits d'accès aux fichiers.

Le basculement d'un ensemble de fichiers de chargement au suivant se fait au niveau du document, même si celui-ci est morcelé à travers plusieurs structures d'analyse communes. Une fois qu'un document a été traité et si un fichier de chargement de l'ensemble des fichiers de chargement dépasse la limite définie, un nouveau fichier de chargement est utilisé. Cela garantit la cohérence de l'ensemble des fichiers de chargement. Une fois le contenu d'un ensemble de fichiers de chargement chargé dans la base de données, le modèle de données reste cohérent car toutes les entrées de la table maîtresse contiennent les entrées corerspondantes dans la table de la base de données.

Les fichiers de chargement et les fichiers de script sont identifiés par l'extension de fichier .cur. Lorsqu'un ensemble de fichiers de chargement est fermé, les fichiers sont renommés pour prendre l'extension .dat. Cela indique que les fichiers peuvent être copiés ou déplacés vers un serveur de base de données tandis que l'analyseur syntaxique de documents s'exécute toujours.

Vous pouvez spécifier la taille d'un fichier de chargement. Lorsque la limite de taille du fichier de chargement est atteinte, un nouvel ensemble de fichiers de chargement est démarré. Vous indiquez la taille de fichier de chargement dans le fichier de mappage de la structure d'analyse commune à la base de données dans la section d'élément XML `<loadFile>`. Le paramètre loadFileSize est défini à l'aide de l'élément `<loadFileSize>` et il est indiqué en mégaoctets avec 10 <= loadFileSize <= 10240 (10 Mo <= loadFileSize <= 10 Go). L'élément `<loadFileSize>` est facultatif. Si aucune valeur n'est définie, la valeur par défaut est 1024 Mo (1 Go).

Les fichiers de chargement uniques d'un ensemble sont numérotés par un nombre à dix chiffres qui identifie quel fichier appartient à quel ensemble de fichiers de chargement. Un ensemble de fichiers de chargement est fermé quand :

- Un fichier de chargement de l'ensemble dépasse la limite de taille définie.
- Le traitement s'est arrêté parce que l'analyseur syntaxique s'est arrêté ou une erreur est survenue.

Si l'analyseur syntaxique est redémarré, le traitement continue de là où il s'était arrêté, en utilisant un nouvel ensemble de fichiers de chargement.

Important : Si vous utilisez Cas2Jdbc pour générer des fichiers de chargement, vérifiez qu'une seule unité d'exécution de l'analyseur syntaxique est configurée. L'utilisation de plusieurs unités d'exécution de l'analyseur syntaxique pour une collection configurée pour générer des fichiers de chargement Cas2Jdbc peut entraîner l'apparition de fichiers de chargement non valides. Pour spécifier le nombre d'unités d'exécution qui seront utilisées, passez par la console d'administration de recherche d'entreprise pour modifier une collection. Sélectionnez la page Analyseur syntaxique, puis l'option permettant de configurer l'analyse syntaxique et spécifiez 1 comme nombre d'unités d'exécution de l'analyseur syntaxique.

Création du fichier de mappage de la structure d'analyse commune à la base de données

Pour ajouter les résultats de l'analyse à une base de données, vous devez créer le fichier de mappage de la structure d'analyse commune à la base de données. Ce dernier contient les informations concernant la configuration de connexion à la base de données et une description des résultats de l'analyse de texte personnalisée à stocker dans des tables et colonnes de la base de données.

A propos de cette tâche

Le fichier de mappage de la structure d'analyse commune à la base de données est en XML. L'exemple suivant repose sur le système de types défini pour le scénario de rapport de police.

Dans l'exemple, seuls les rapports de polices et les villes sont ajoutés à la base de données. L'exemple présente l'utilisation de fonctions intégrées et du mappage de l'élément <constant>.

```
<?xml version="1.0" encoding="UTF-8"?>
<cas2JdbcConfiguration xmlns="http://www.ibm.com/uima/consumer/jdbc/100/xml">
   <databaseConnection>
     <connectionUrl>db2://myMachine:myPort/myDatabase</connectionUrl>
     <driver type="jdbc">com.ibm.db2.jcc.DB2Driver</driver>

     <driverLibraries>
       <driverLibrary>C:\db2\db2jcc.jar</driverLibrary>
       <driverLibrary>C:\db2\db2jcc_license_cu.jar</driverLibrary>
       <driverLibrary>C:\db2\db2jcc_license_cisuz.jar</driverLibrary>
     </driverLibraries>

     <authentication>
       <username>myUser</username>
       <password>myPassword</password>
     </authentication>

     <loadFile>
       <loadFileDirectory>/home/cas2jdbc/load/</loadFileDirectory>
       <loadFileSize>1048</loadFileSize>
       <loadScript>/home/cas2jdbc/load/load.sh</loadScript>
     </loadFile>

   </databaseConnection>

   <cas2JdbcMappingSpec>
     <skipCondition>
       <name>com.ibm.uima.tt.DocumentAnnotation</name>
       <filter syntax="FeatureValue">toBeProcessed=0</filter>
     </skipCondition>

     <cas2JdbcMappings>
      <explicitMappings>
       <explicitMappingRule applyToSubtypes="false">
        <type>com.ibm.omnifind.types.PoliceReport</type>
        <table>sample.policeReport</table>
        <featureMappings>
         <featureMapping>
          <feature>uniqueId()</feature>
          <column>policeReportId</column>
         </featureMapping>
         <featureMapping>
          <feature>location/uniqueId()</feature>
          <column>crimeLocationId</column>
         </featureMapping>
        </featureMappings>
        <filter syntax="FeatureValue">location/coveredText()="Los Angeles"</filter>
       </explicitMappingRule>
      </explicitMappings>

      <implicitMappings>
       <implicitMappingRule applyToSubtypes="false">
        <type>com.ibm.omnifind.types.City</type>
         <table>sample.City</table>
        <featureMappings>
         <featureMapping>
          <feature>uniqueId()</feature>
          <column>crimeLocationId</column>
```

```
        </featureMapping>
        <featureMapping>
         <feature>coveredText()</feature>
         <column>cityName</column>
         <length>150</length>
        </featureMapping>
        <featureMapping>
         <constant>USA</constant>
         <column>country</column>
        </featureMapping>
       </featureMappings>
      </implicitMappingRule>
     </implicitMappings>

    </cas2JdbcMappings>
  </cas2JdbcMappingSpec>
</cas2JdbcConfiguration>
```

Procédure

Pour créer le fichier de mappage de la structure d'analyse commune à la base de données :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix. Le schéma XSD du fichier de mappage est appelé `CasToJDBCMapping.xsd` et se trouve dans votre installation de recherche d'entreprise à l'emplacement *RACINE_INSTALL_RECHERCHE_ENTREPRISE*`/packages/uima/configuration_xsd/`.
2. Incluez vos mappages dans un élément `<cas2JdbcConfiguration xmlns="http://www.ibm.com/uima/consumer/jdbc/100/xml">`. L'espace de nom (indiqué dans l'attribut xmlns) doit être exactement identique à celui affiché.
3. Ajoutez un élément `<databaseConnection>` qui contient toutes les informations de configuration de connexion à une base de données et un élément `<cas2JdbcMappingSpec>` qui décrit les règles de mappage pour les résultats de l'analyse qui sont stockés dans la base de données ou dans des fichiers de chargement.
4. Ajoutez les éléments de composant suivants à l'élément `<databaseConnection>` :
   - Obligatoire : Un élément `<connectionUrl>`. Cet élément contient l'URL de connexion à la base de données. En fonction de l'implémentation du pilote JDBC, vous pouvez utiliser un accès distant ou local à la base de données.
   - Obligatoire : Un élément `<driver>`. Cet élément contient le nom de la classe du pilote JDBC, par exemple `com.ibm.db2.jcc.DB2Driver` pour DB2, ou `oracle.jdbc.driver.OracleDriver` pour Oracle.
   - Obligatoire : Un élément `<driverLibraries>`. Cet élément permet d'établir une liste des bibliothèques du pilote. Chaque bibliothèque est répertoriée dans un élément `<driverLibrary>`. Les bibliothèques se trouvent dans le répertoire d'installation DB2 ou Oracle. Pour DB2, les bibliothèques sont `c:\your_db2_dir\db2jcc.jar`, `c:\your_db2_dir\db2jcc_license_cu.jar` et `c:\your_db2_dir\db2jcc_license_cisuz.jar`. Pour Oracle, la bibliothèque à inclure est `c:\votre_rép_oracle\classes12.zip`.

     Vérifiez que les bibliothèques du pilote sont toujours au même niveau de maintenance que le serveur d'applet DB2.
   - Obligatoire : Un élément `<authentication>`. Cet élément contient le nom d'utilisateur et le mot de passe pour la base de données.
   - Facultatif : Un élément `<loadFile>`. Cet élément contient les éléments de composants suivants :

- Le répertoire du fichier de chargement est un élément <loadFileDirectory>.
- Facultatif : La taille du fichier de chargement dans un élément <loadFileSize>. Les limites de taille du fichier de chargement sont de 10 <= loadFileSize <= 10240 (10 Mo <= loadFileSize <= 10 Go). Si aucune valeur n'est définie, la valeur par défaut est de 1024 Mo (1 Go).
- Le nom de script de chargement dans un élément <loadScript>.

Si vous n'indiquez pas d'élément <loadFile>, toutes les données sont stockées directement dans la base de données à l'aide de JDBC.

Vous devez également ajouter l'ensemble des paramètres de configuration de base de données lorsque vous utilisez des fichiers de chargement et des scripts propres à la base de données.

5. Ajoutez les éléments de composant suivants à l'élément <jdbcMappingSpec> :
   - Facultatif : Un élément <skipCondition>. Si aucune condition skip n'est définie, tous les documents sont traités.

     ```
     <skipCondition>
       <name>com.ibm.uima.tt.DocumentAnnotation</name>
       <filter syntax="FeatureValue">toBeProcessed=0</filter>
     </skipCondition>
     ```

     Dans l'exemple, les documents qui contiennent une annotation de type com.ibm.uima.tt.DocumentAnnotation avec une fonction nommée toBeProcessed dont la valeur est égale à zéro ne sont pas pris en compte.
   - Un élément <cas2JdbcMappings> qui indique les types et les fonctions mappés à des tables et à des colonnes de base de données spécifiques. L'élément contient une section de mappages explicites et implicites.
6. Ajoutez un élément <explicitMappings>. Cet élément est obligatoire. Il doit avoir un ou plusieurs éléments <explicitMappingRule> qui définissent les mappages explicites et peut être défini uniquement pour les types d'annotation et leurs sous-types. Si un mappage est défini dans une section de mappages explicites, toutes les annotations qui correspondent à la définition de mappage sont stockées dans la base de données.
7. Facultatif : Ajoutez un élément <implicitMappings>. Cet élément prend en charge tous les types de structure de fonctions. Si cet élément existe, il doit contenir au moins un élément <implicitMappingRule>. Les mappages définis dans la section des mappages implicites sont ajoutés à la base de données uniquement si les types d'annotation sont référencés par une autre annotation qui respecte une règle de mappage implicite ou explicite.

   Le but du mappage implicite est de permettre le stockage uniquement des résultats de l'analyse qui apparaissent dans un contexte particulier. Par exemple, si le mappage d'une annotation de type com.ibm.omnifind.types.City est implicite, seules les villes référencées par la définition de mappage com.ibm.omnifind.types.PoliceReport de la section de mappages explicites sont stockées dans la base de données. Autrement dit, seules les villes mentionnées dans des rapports de police sont ajoutées à la base de données.

   S'il existe une règle de mappage explicite pour l'annotation City, toutes les villes sont ajoutées à la base de données. Dans tous les cas, si une ville est référencée par plusieurs rapports de police, elle est ajoutée une seule fois à la base de données.
8. Les éléments <explicitMappingRule> et <implicitMappingRule> doivent contenir l'attribut applyToSubtypes qui lorsqu'il a la valeur true stocke non seulement la structure de la fonction indiquée dans l'élément <type> mais

également toutes les structures qui en sont dérivées. Ajoutez les éléments de composant suivants aux éléments <explicitMappingRule> et <implicitMappingRule> :

- Un élément <type> qui contient le type de structure de fonctions.
- Un élément <table> qui contient le schéma de base de données et le nom de la table. La syntaxe suit la règle schema.table_name, ou uniquement table_name si aucun schéma n'est défini.
- Un élément <featureMappings> avec un ou plusieurs éléments <featureMapping> ou un élément <containerMapping>.
- Facultatif : Un élément <filter> qui contient une condition évaluée dès concordance de la règle de mappage. Si la condition a la valeur true, l'annotation ou la structure de fonctions est stockée dans la base de données. Dans cet exemple, seuls les rapports de police concernant des crimes commis à Los Angeles seront stockés dans la base de données.

9. La structure de composant de l'élément <featureMapping> diffère selon que vous mappez une fonction ou une constante.

   Si vous mappez une fonction ou un chemin de fonctions, les éléments du composant incluent :

   - Un élément <feature> avec le nom de la fonction. La fonction doit être définie pour la structure de la fonction dans l'élément type. Vous pouvez également utiliser une construction de chemin de fonctions ou une des fonctions intégrées du système.
   - Facultatif : Un élément <length> avec la longueur d'une chaîne admise dans la colonne de base de données indiquée. Les chaînes plus longues sont tronquées.
   - Un élément <column> avec le nom de la colonne dans laquelle la valeur de la fonction doit être stockée. Les colonnes de base de données qui ne sont pas utilisées dans des mappages de fonction utilisent une valeur par défaut (généralement, la valeur null) qui est configurée dans la base de données.

     Assurez-vous que la valeur de l'élément feature est stockée dans une colonne du type approprié. Le tableau suivant indique quels types UIMA correspondent à quels types de base de données.

Tableau 3. Mappage entre les types UIMA et les types de base de données correspondants

Type UIMA ou fonction intégrée	**Type de données DB2 recommandé**	**Type de données Oracle recommandé**
Float	REAL	FLOAT
String	VARCHAR	VARCHAR2
Integer	INTEGER	INTEGER
uniqueId(), uniqueParentId()	CHAR(27)	CHAR(27)
objectId(), parentId()	CHAR(16)	CHAR(16)
docTimestamp()	BIGINT	LONG
fsId()	INTEGER	INTEGER

   Pour une constante, les éléments de mappage de fonction de composant sont les suivants :

   - Un élément <constant> qui contient la valeur d'une constante.
   - Un élément <column> avec le nom de la colonne dans lequel la valeur de la constante est ajoutée.

10. L'élément <containerMapping> contient le mappage d'une fonction de type de conteneur (tableau ou liste). Cet élément doit être utilisé uniquement pour les types de conteneur. Il comporte les éléments de composant suivants :
    - Un élément <feature> avec le nom de la fonction. Vous pouvez également utiliser une construction de chemin de fonctions ou une des fonctions intégrées du système.
    - Un élément <table> qui contient le schéma de base de données et le nom de la table. La syntaxe suit la règle schema.table_name, ou uniquement table_name si aucun schéma n'est défini.
    - Un ou plusieurs éléments <featureMapping> qui contiennent le nom des structures de fonction et le noms des colonnes dans lesquelles les fonctions sont ajoutées.
11. Sauvegardez et validez le fichier XML à l'aide du schéma fourni.

Une fois que vous avez créé le fichier XML, vous devez le télécharger dans la recherche d'entreprise et sélectionner le fichier de mappage de la structure d'analyse commune à la base de données ainsi que d'autres options d'analyse personnalisées dans la console d'administration de la recherche d'entreprise.

Concepts associés

«Mappage de base de données pour les résultats de l'analyse sélectionnés», à la page 47

«Chemins de fonctions», à la page 35

Référence associée

«Filtres», à la page 39

«Fonctions intégrées», à la page 37

«Exemple de description de système de types», à la page 25

Mappage de type de conteneur

Un type de conteneur est un des types de liste ou de tableau intégré de la structure d'analyse commune. Le mappage de type de conteneur permet de mapper des valeurs de liste ou de tableau à une base de données relationnelle.

Vous disposez de deux approches pour la gestion des types de conteneur dans le fichier de mappage de la structure d'analyse commune à la base de données. Une méthode utilise des constructions de fonctions intégrées et une table de liens génériques incluant des tableaux ou des listes qui constituent les valeurs d'une règle de mappage de fonction. Etant donné que différents tableaux ou listes sont stockés dans la même table de liens, cette dernière ne comporte aucune informations sur la relation des informations stockées.

Dans la deuxième méthode, la définition de table de liens générée à l'aide de l'élément <containerMapping> indique explicitement la relation entre les informations indiquées requises.

Un exemple de mappage de table de liens générique est disponible ci-dessous. Il existe une relation n:m entre les rapports de police et les suspects. Ce qui signifie qu'un suspect peut être mentionné dans plusieurs rapports de police et qu'un rapport de police peut mentionner plusieurs suspects.

La table sample.fsarray générique de l'exemple constitue un lien entre les rapports de police et les suspects. S'il existe un type de mappage autre que com.ibm.omnifind.types.PoliceReport qui comporte une fonction de type com.ibm.omnifind.types.FSArray, il est également mappé à cette table. Vous

pouvez toujours interroger la table pour connaître la relation entre un rapport de police et un suspect. Toutefois, vous ne pouvez pas conclure en consultant uniquement la table que cette dernière contient la relation ou le lien entre les rapports de police et les suspects possibles.

```
<cas2JdbcMappings>
  <explicitMappings>
    <explicitMappingRule applyToSubtypes="false">
      <type>com.ibm.omnifind.types.PoliceReport</type>
      <table>sample.policeReport</table>
      <featureMappings>
        <featureMapping>
          <feature>uniqueId()</feature>
          <column>policeReportId</column>
        </featureMapping>
        <featureMapping>
          <feature>knownSuspects/uniqueId()</feature>
          <column>suspectArrayId</column>
        </featureMapping>
        <featureMapping>
          <feature>location/cityName</feature>
          <column>city</column>
        </featureMapping>
      </featureMappings>
    </explicitMappingRule>
  </explicitMappings>

  <implicitMappings>
    <implicitMappingRule applyToSubtypes="false">
      <type>com.ibm.omnifind.types.Suspect</type>
      <table>sample.suspect</table>
      <featureMappings>
        <featureMapping>
          <feature>uniqueId()</feature>
          <column>suspectID</column>
        </featureMapping>
        <featureMapping>
          <feature>surName</feature>
          <column>lastName</column>
        </featureMapping>
        <featureMapping>
          <feature>description</feature>
          <column>description</column>
        </featureMapping>
      </featureMappings>
    </implicitMappingRule>

    <implicitMappingRule applyToSubtypes="false">
      <type>uima.cas.FSArray</type>
      <table>sample.fsarray</table>
      <featureMappings>
        <featureMapping>
          <feature>uniqueId()</feature>
          <column>arrayId</column>
        </featureMapping>
        <featureMapping>
          <feature>[:index]</feature>
          <column>arrayIndex</column>
        </featureMapping>
        <featureMapping>
          <feature>[]/uniqueId()</feature>
          <column>suspectId</column>
        </featureMapping>
      </featureMappings>
```

```
    </implicitMappingRule>
  </implicitMappings>

</cas2JdbcMappings>
```

L'élément suivant affiche les tables de base de données créées à l'aide des règles de mappage génériques ci-dessus.

Tableau 4. Table sample.policeReport

policeReportId	**suspectArrayId**	**city**
aaa...1	**bbb...1**	Springfield
aaa...2	**bbb...2**	Ladysmith

Tableau 5. Table sample.fsarray

arrayId	**arrayIndex**	**suspectId**
bbb...1	1	*ccc...1*
bbb...1	2	*ccc...2*
bbb...2	1	*ccc...3*

Tableau 6. Table sample.suspect

suspectID	**lastname**	**description**
ccc...1	Brown	Dark complexion
ccc...2	Smith	Wears glasses
...	...	...

L'exemple affiche le mappage des tableaux de structures de fonctions. Vous pouvez également appliquer ce type de mappage à StringArray, IntegerArray et FloatArray. Si vous incluez des règles de mappage pour ces tableaux de valeurs simples, remplacez `[]/uniqueId()` par `[]`.

L'approche de table générique peut être utilisée pour les listes de structures de fonctions ainsi que pour les listes de types simples (StringList, IntegerList et FloatList).

Une méthode plus simple de gestion des relations consiste à utiliser un élément de mappage de conteneur explicite qui définit l'itération des éléments se trouvant dans les tableaux ou les listes.

Un exemple de mappage qui décrit une table de liens explicites apparaît ci-dessous. Il existe à nouveau une relation n:m entre les rapports de police et les suspects. Toutefois, ici la table `sample.reports_suspects` constitue la table de liens entre les rapports de police et les suspects.

En utilisant cette approche, vous n'avez pas à prendre en compte les ID de tableau ou les mappages d'entrée de début et de fin pour les types de liste. La table de liens contient une relation explicite.

```
<cas2JdbcMappings>
  <explicitMappings>
    <explicitMappingRule applyToSubtypes="false">
      <type>com.ibm.omnifind.types.PoliceReport</type>
      <table>sample.policeReport</table>
      <featureMappings>
```

```
        <featureMapping>
          <feature>uniqueId()</feature>
          <column>policeReportID</column>
        </featureMapping>
        <featureMapping>
          <feature>location/cityName</feature>
          <column>city</column>
        </featureMapping>
        <featureMapping>
          <feature>knownSuspects</feature>
          <containerMapping>
            <table>sample.reports_suspects</table>
            <featureMapping>
              <feature>com.ibm.omnifind.types.PoliceReport
                /objectId()</feature>
              <column>policeReportId</column>
            </featureMapping>
            <featureMapping>
              <feature>knownSuspects/[]/objectId()</feature>
              <column>suspectId</column>
            </featureMapping>
          </containerMapping>
        </featureMapping>
      </featureMappings>
    </explicitMappingRule>
  </explicitMappings>

  <implicitMappings>
    <implicitMappingRule applyToSubtypes="false">
      <type>com.ibm.omnifind.types.Suspect</type>
      <table>sample.suspect</table>
      <featureMappings>
        <featureMapping>
          <feature>objectId()</feature>
          <column>suspectID</column>
        </featureMapping>
        <featureMapping>
          <feature>surName</feature>
          <column>lastName</column>
        </featureMapping>
        <featureMapping>
          <feature>description</feature>
          <column>description</column>
        </featureMapping>
      </featureMappings>
    </implicitMappingRule>
  </implicitMappings>

</cas2JdbcMappings>
```

Un élément `<containerMapping>` permet de définir les itérations des éléments contenus dans le tableau. Dans l'exemple, la table de liens `sample.reports_suspects` contient un lien vers les colonnes `policeReportId` et `suspectId`. N'imbriquez pas les éléments `<containerMapping>`.

L'élément suivant affiche les tables de base de données créées à l'aide de règles de mappage de tables de liens explicites.

Tableau 7. Table sample.policeReport

policeReportId	**city**
aaa...1	Springfield
aaa...2	Ladysmith

Tableau 8. Table sample.reports_suspect

policeReportId	**suspectId**
bbb...1	*ccc...1*
bbb...2	*ccc...2*
...	...

Tableau 9. Table sample.suspect

suspectID	**lastname**	**description**
ccc...1	Brown	Dark complexion
ccc...2	Smith	Wears glasses
...	...	...

Référence associée

«Fonctions intégrées», à la page 37

Extraction des parties d'un document qui correspondent à une requête de recherche sémantique

Vous pouvez extraire uniquement les parties d'un document qui correspondent exactement à la requête en mappant les structures de fonctions appropriées à l'index et à la base de données et en indiquant l'étendue dans la requête de recherche sémantique.

Pour accéder à toutes les instances d'un type d'annotation spécifique dans le résultat de la recherche, par exemple, pour obtenir toutes les personnes, incluez un mappage de style de zones pour le type d'annotation et indiquez qu'il peut être renvoyé dans le fichier de mappage de la structure d'analyse commune à l'index. Par exemple :

```
<indexBuildItem>
 <name>com.ibm.omnifind.types.Person</name>
 <indexRule>
  <style name="Annotation"/>
  <style name="Field">
   <attribute name="returnable" value="true"/>
  </style>
 </indexRule>
</indexBuildItem>
```

Dans cet exemple, les annotations de type `com.ibm.omnifind.types.Person` sont mappées à une étendue nommée Person dans l'index de recherche d'entreprise. Vous pouvez utiliser ce dernier pour accéder aux annotations lors de la recherche sémantique. De plus, le texte couvert des annotations (des noms de personnes, par exemple) est stocké sous la forme de zone pouvant être envoyée. Pour extraire ces valeurs d'annotation, appelez getFields("Person") sur chaque objet de résultat renvoyé de la requête de recherche (mot clé ou sémantique). Cette méthode renvoie un tableau de chaînes avec les valeurs d'annotation. Dans le cas présent, le noms des personnes.

Toutefois, cette approche renvoie toutes les instances d'une annotation donnée et n'est pas appropriée si vous souhaitez limiter le traitement du résultat aux documents qui correspondent exactement à la requête. Par exemple, un document peut mentionner cinq personnes. Toutefois, dans la requête de recherche sémantique '`<sentence><person/>IBM</sentence>`, l'utilisateur est intéressé

uniquement par la personne mentionnée dans la même phrase que celle dans laquelle apparaît dans le terme IBM. Les autres personnes ne l'intéressent pas.

Pour accéder et traiter les structures de fonctions qui correspondent exactement à la requête, procédez comme suit :

1. Mappez les types de structure de fonctions appropriés à l'index de recherche d'entreprise à l'aide du style de mappage d'annotation. Par exemple :

```
<indexBuildItem>
 <name>com.ibm.omnifind.types.Person</name>
 <indexRule>
  <style name="Annotation"/>
 </indexRule>
</indexBuildItem>
```

2. Mappez les types de structure de fonctions appropriés aux tables JDBC. Au cours du processus de mappage, vous devez inclure deux colonnes, une pour l'URI de document et l'autre pour l'ID de structure de fonctions. Bien que vous puissiez mapper l'ensemble des types de structure de fonctions à la même table de base données, vous devez mapper chaque type à une table différente. Par exemple :

```
<explicitMappingRule applyToSubtypes="false">
 <type>com.ibm.omnifind.types.Person</type>
 <table>sample.person</table>
 <featureMappings>
  <featureMapping>
   <feature>objectId()</feature>
   <column>primaryId</column>
  </featureMapping>
  <!-- Contient le texte couvert de l'annotation-->
  <featureMapping>
   <feature>coveredText()</feature>
   <column>personName</column>
  </featureMapping>
  <!-- Autre mappage ici-->
  <!-- Pour accéder aux annotations de personnes pertinentes
                    dans le résultat de requête-->
  <featureMapping>
   <feature>docUri()</feature>
   <column>docUri</column>
  </featureMapping>
  <featureMapping>
   <feature>fsId()</feature>
   <column>annotationId</column>
  </featureMapping>
 </featureMappings>
</explicitMappingRule>
```

3. Parcourez, analysez et indexez les documents.
4. Extrayez les ID des instances correspondant à la requête. Dans l'interface SIAPI (search and index API), ces éléments sont appelés éléments cibles. Un élément cible indique l'étendue cible à renvoyer. Il est défini de la manière suivante :
   - Dans les fragments XML, l'élément cible est identifié par un signe numéro (#). Ce signe est autorisé uniquement une fois et peut apparaître à tout emplacement de la requête de fragment XML. Par exemple : `$xmlf2::'<sentence><#person/>IBM</sentence>'`
   - Dans XPath par défaut, l'élément cible est la dernière zone de l'expression XPath.
   - Accédez à ces instances en utilisant la méthode `Result.getProperty("TargetElement")`. La propriété renvoyée est une concaténation de chaînes de tous les ID d'occurrence séparés par des espaces. Chaque occurrence de la propriété peut être convertie en une valeur entière.

5. SIAPI ne renvoie pas les structures de fonctions elles-mêmes, uniquement leurs ID d'occurrence. Ces ID correspondent à la valeur fsId() stockée dans la table de base de données. Pour extraire ces instances et les informations associées, votre application doit :
   a. sélectionner la table de base de données correcte, en fonction du nom d'étendue de l'élément cible. Dans l'exemple, l'application contient un mappage d'une personne à la table sample.Person. Ces informations sont déduites du fichier de mappage de la structure d'analyse commune à l'index, qui renvoie le nom d'étendue et du fichier de mappage de la structure d'analyse commune à la base de données, qui renvoie le nom de table.
   b. Pour chaque résultat de la recherche :
      1) Analysez la chaîne renvoyée par `Result.getProperty ("TargetElement")` pour trouver les ID d'occurrence.
      2) Emettez une instruction SELECT pour la table en utilisant l'URI de résultat (accessible à l'aide de `Result.getDocumentId()`) en tant que valeur de la colonne docUri et les ID d'occurrence en tant que valeur de colonne annotationId. Les noms de colonne dépendent du fichier de configuration. Les noms de colonne utilisés proviennent de l'exemple précédent.

      Les lignes renvoyées contiennent les informations stockées pour la structure de fonctions, par exemple, le texte couvert ou les attributs spécifiques de la structure de fonctions, telle que "last name" ou "city of birth."

Vérifiez que les mises à jour apportées à la base de données sont synchronisées aux mises à jour d'index dans la recherche d'entreprise. Si la base de données contient des informations obsolètes (par exemple, vous avez utilisé des fichiers de chargement de base de données et vous n'avez pas mis à jour cette dernière mais vous avez régénéré ou réorganisé l'index), certains ID d'occurrence peuvent ne pas se trouver dans la base de données. La recherche d'entreprise conserve un enregistrement de la dernière version du document dans son index. C'est pourquoi, les ID d'occurrence sont valides uniquement pour le dernier document.

Si vous stockez plusieurs versions du même document dans la même table de base de données, plusieurs lignes peuvent correspondre aux mêmes ID d'occurrence, pour les différentes versions du document. Vous devez alors définir une colonne de version de document et la charger en utilisant la logique de l'application ou les fonctions intégrées, telles docTimestamp(). Ainsi, vous pouvez filtrer le résultat afin d'obtenir uniquement la dernière version du document.

Concepts associés

«Terme de requête de recherche sémantique», à la page 61

Tâches associées

«Création du fichier de mappage de la structure d'analyse commune à l'index», à la page 41

«Création du fichier de mappage de la structure d'analyse commune à la base de données», à la page 49

Applications de recherche sémantique

Quatre types d'informations de document sont stockés dans l'index de recherche d'entreprise que vous pouvez interroger dans les applications de recherche à l'aide de l'interface SIAPI.

Les quatre différents types d'informations incluent :

- Des mots qui se trouvent dans un document, par exemple une chaîne telle *logiciel*.
- Des noms d'étendue, par exemple, un document XML qui inclut <author>James</author> génère l'étendue <author>.
- Des noms d'attributs, par exemple un document XML qui inclut <author countryOfBirth=USA>James</author>, génère l'attribut "countryOfBirth".
- Des valeurs d'attribut, par exemple USA est la valeur de l'attribut "countryOfBirth."

La langue de la requête SIAPI inclut le terme de la requête de recherche sémantique. Le terme spécifie un motif de brindille. Une brindille est un petit arbre avec des feuilles. Chaque feuille représente les quatre types d'informations (mots, noms d'étendue, etc). Les modes internes de l'arborescence indiquent comment les occurrences d'un document sont liées les unes aux autres. Il existe cinq types de noeuds internes qui définissent des relations :

- et
- ou
- non
- dans_étendue_de
- attribut_dans_portée_de

Un document satisfait une recherche sémantique donnée s'il inclut les occurrences des feuilles et que les contraintes définies par les noeuds internes (relations définies) sont respectées.

La requête de recherche sémantique vous aide à extraire des documents de meilleure qualité. Maintenant, vous pouvez non seulement effectuer une recherche en utilisant des combinaisons booléennes de mots et d'annotations mais vous pouvez également extraire des document dans lesquels, par exemple *James* apparaît dans l'étendue nommée author ou dans lesquels les termes *ibm* et *search* apparaissent dans la même phrase.

Terme de requête de recherche sémantique

Le terme de requête de recherche sémantique est communiqué sous la forme d'un terme opaque.

Il existe deux formes de syntaxe permettant d'exprimer un terme opaque dans l'interface SIAPI :

- Fragments XML
- XPath limité

Le terme de requête de fragment XML a l'aspect d'un fragment équilibré d'un document XML. Un terme de requête de fragment XML est préfixé par le signe de terme opaque `@xmlf2::` suivi de l'expression de fragment XML incluse entre apostrophes ('...').

Toutefois, les termes de requête XPath limités sont préfixés par `@xmlxp::` suivi de la requête XPath incluse entre apostrophes ('...').

De la même manière qu'avec des termes de requête généraux de l'interface SIAPI, chaque terme peut avoir un modificateur d'apparence :

Signe plus (+)
Le terme doit apparaître.

Préfixe =
Le terme doit être une correspondance exacte.

Caractère tilde (~) en préfixe
Prise en compte du terme de la requête.

Caractère tilde (~) en suffixe
Prise en compte des mots qui ont le même lemme que le terme de la requête.

Signe dièse (#)
Le terme est mis en évidence.

Les exemples suivants affichent des requêtes de fragment XML.

@xmlf2::'<City>Springfield</City>'
Recherche des documents qui incluent l'étendue (annotation) City contenant la chaîne Springfield.

@xmlf2::'<Person gender="female"/>'
Recherche des documents dans lesquels une personne de sexe féminin est annotée.

@xmlf2::'<Person><.or><@gender>female</@gender> <@title>Mrs</@title><@title>Ms</@title></.or></Person>'
Recherche des documents qui définissent une personne en tant que femme à l'aide du sexe ou de la civilité.

@xmlf2::'<Person gender="male" role="suspect"/> <PoliceReport><@crimeDescription><.or>robbery theft</.or>-accident </@crimeDescription></PoliceReport> <City>Springfield<.or> <@district>Brynston</@district><@district>Brooklyn</@district><.or></City>'
Recherche des documents qui indiquent des individus de sexe masculin considérés comme suspects et une annotation PoliceReport attribuée par la chaîne *robbery* ou *theft* dans l'attribut crimeDescription, mais pas la chaîne *accident*. Les documents doivent également contenir une annotation city couvrant le mot du texte *Springfield*, une annotation qui est attribuée au district *Brynston* ou *Brooklyn*.

Les requêtes XPath correspondantes ont les structures suivantes :

@xmlxp::'//City ftcontains ("Springfield")'
Recherche des documents qui incluent l'étendue (annotation) City contenant la chaîne *Springfield*.

@xmlxp::'//PoliceReport[City ftcontains("Springfield")]'
Recherche des documents qui incluent l'étendue (annotation) City dans l'étendue PoliceReport contenant la chaîne *Springfield*.

@xmlxp::'//Person[@gender="female" or @title ftcontains("Ms") or @title ftcontains("Mrs")]'
Recherche des documents dans lesquels une personne de sexe féminin est

annotée. Dans l'attribut gender, la valeur doit correspondre exactement, tandis que pour l'attribut title, *Ms* et *Mrs* n'ont pas besoin de correspondre exactement à la valeur d'attribut.

Prise en charge des synonymes dans les applications de recherche

Vous pouvez étendre les résultats de la recherche en recherchant des documents contenant des synonymes de termes de la requête.

Les synonymes comprennent généralement des termes incluant plusieurs mots, des noms de produit par exemple (tel *OmniFind Enterprise Edition*). Les termes incluant plusieurs mots contenus dans le dictionnaire de synonymes sont correctement identifiés dans les requêtes utilisateur et ne sont pas placés entre guillemets.

L'API SIAPI (Search and Index API) pour la recherche d'entreprise prend en charge plusieurs méthodes de recherche de synonymes des termes de la requête :

- La syntaxe des requêtes SIAPI prend en charge l'opérateur tilde (~) pour le développement des synonymes. Si l'utilisateur ajoute cet opérateur à un terme de la requête, le développement de synonymes est effectué pour ce mot. Par exemple, la requête `~WAS` renvoie des documents qui comportent WebSphere Application Server et tout autre synonyme de cette abréviation.
- Le développement de synonymes peut être activé à l'aide de l'interface d'expansion de synonymes SIAPI à partir d'une application de recherche. Les termes de la requête peuvent être automatiquement développés afin d'inclure des synonymes ou l'application de recherche peut inclure des options qui permettent à l'utilisateur de déterminer si les synonymes des termes de la requête doivent être renvoyés dans les résultats de la recherche.

  Lors du développement automatique de synonymes, la recherche de synonymes est effectuée sur tous les mots de la requête. Les résultats de la recherche incluent des documents qui contiennent, soit les termes de la requête, soit des synonymes des termes de la requête. La SIAPI prend également en charge la génération d'une liste d'extensions de synonymes pour la requête soumise.
- Le développement de synonymes dans les collections n-gram permet la segmentation des phrases du texte de la requête. Si une phrase entière apparaît dans le dictionnaire de synonymes, cela signifie que la recherche a abouti. Une phrase est extraite en fonction des délimiteurs suivants :

  Ponctuation

  Les caractères suivants sont des délimiteurs : `- ( ) + . ,`

  Les guillemets sont ignorés et ne séparent pas les phrases.

  Modification de l'alphabet

  Par exemple, pour une collection n-gram, la requête sera développée pour inclure les synonymes de `ABC` dans les exemples de requête suivants si `ABC` se trouve dans le dictionnaire de synonymes :

  `ABC run DEF stand` (où `ABC` et `DEF` sont en japonais)
  `ABC+DCF+GHI`

Création d'un fichier XML pour les synonymes

Pour développer des requêtes dans la recherche d'entreprise afin d'inclure les synonymes des termes de la requête, vous devez indiquer quels mots sont synonymes de quels autres dans un fichier XML. Ce fichier XML est utilisé pour générer un fichier de dictionnaire binaire que vous téléchargez vers la recherche d'entreprise et attribuez aux collections appropriées.

A propos de cette tâche

Le fichier XML qui répertorie les synonymes doit se conformer à un schéma spécifique. Voici un exemple de fichier XML pour les synonymes :

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">
  <synonymgroup>
    <synonym>Think Pad</synonym>
    <synonym>Notebook</synonym>
    <synonym>Notebooks</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>WebSphere Application Server</synonym>
    <synonym>WAS</synonym>
  </synonymgroup>
</synonymgroups>
```

Restrictions

Vous devez grouper des mots qui sont synonymes les uns des autres (éléments `<synonym>`) dans un élément `<synonymgroup>`. Un synonyme peut inclure des espaces mais ne peut pas inclure de caractères de ponctuation, tels une virgule (,) ou une barre verticale (|), car ces caractères peuvent entrer en conflit avec la syntaxe de requête de recherche d'entreprise.

Vous devez énumérer toutes les déclinaisons possibles des termes ajoutés en tant que synonymes, telles la forme au singulier et au pluriel d'un mot. Il n'est pas nécessaire d'énumérer les normalisations du terme, comme la suppression des accents ou des trémas allemands (Umlaut). La recherche d'entreprise gère automatiquement la normalisation. Il n'est pas utile non plus d'inclure les variantes du terme en majuscules et en minuscules. Par exemple, si vous souhaitez inclure le terme météo en tant que synonyme, il n'est pas nécessaire d'inclure également le terme METEO.

Procédure

Pour créer une liste de synonymes pour la recherche d'entreprise, procédez comme suit :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix. Le schéma XSD du fichier XML est appelé `synonyms.xsd` et se trouve dans votre installation de recherche d'entreprise à l'emplacement *RACINE_INSTALL_RECHERCHE_ENTREPRISE*`/packages/uima/configuration_xsd/`.
2. Ajoutez un élément `<synonymgroup>`, puis un élément `<synonym>` pour chaque mot devant être traité comme un synonyme d'autres mots dans le groupe de synonymes.

   Vérifiez que vos mappages sont inclus dans un élément `<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">`. L'espace de nom (indiqué dans l'attribut xmlns) doit être exactement identique à celui affiché.

3. Répétez la procédure jusqu'à ce que vous ayez spécifié tous les synonymes à utiliser pour la recherche de documents dans une collection de recherche d'entreprise.
4. Sauvegardez et quittez le fichier XML.

Une fois que vous avez créé le fichier XML, vous devez le convertir en un dictionnaire de synonymes afin qu'il puisse être ajouté au système de recherche d'entreprise.

Création d'un dictionnaire de synonymes

Une fois que vous avez créé ou mis à jour une liste de synonymes dans un fichier XML, vous devez convertir ce dernier en un dictionnaire de synonymes binaire.

A propos de cette tâche

Pour créer un dictionnaire de synonymes, utilisez l'outil de ligne de commande appelé `essyndictbuilder`, fourni avec OmniFind Enterprise Edition. Il se trouve dans le répertoire *ES_INSTALL_ROOT*`/bin`.

L'outil traite un fichier XML qui répertorie les synonymes et génère un dictionnaire de synonymes. Le dictionnaire doit avoir le suffixe `.dic`. Par exemple, `c:\mydictionaries\products.dic`.

L'emplacement par défaut des fichiers est le répertoire à partir duquel le script est appelé. S'il existe un dictionnaire portant le même nom, le script génère une erreur.

La taille maximale d'un `.dic` dans la recherche d'entreprise est de 8 Mo.

Procédure

Pour créer un dictionnaire de synonymes pour la recherche d'entreprise, procédez comme suit :

1. Sur le serveur d'index, connectez-vous en tant qu'administrateur de recherche d'entreprise. Cet ID utilisateur a été indiqué lors de l'installation de OmniFind Enterprise Edition.
2. Entrez la commande suivante, où *fichier_XML* correspond au chemin complet du fichier XML qui contient la liste des synonymes et *fichier_DIC* au chemin complet du dictionnaire de synonymes.

   AIX, Linux ou Solaris : `essyndictbuilder.sh` *fichier_XML fichier_DIC*
   Windows : `essyndictbuilder.bat` *fichier_XML fichier_DIC*

Une fois que vous avez créé un dictionnaire de synonymes, utilisez la console d'administration de recherche d'entreprise pour ajouter le dictionnaire au système de recherche d'entreprise et l'associer à une ou à plusieurs collections.

Seul le fichier `.dic` généré est téléchargé vers le système de recherche d'entreprise. Assurez-vous que le fichier XML source est stocké dans un environnement dont l'accès est contrôlé et effectuez une sauvegarde régulière du fichier. Ce fichier XML est requis pour la mise à jour du dictionnaire de synonymes.

Dictionnaires de mots vides personnalisés

Vous pouvez définir un vocabulaire propre à l'entreprise qui est retiré d'une requête afin d'augmenter la pertinence de la recherche.

Il existe deux types de prise en charge de mots vides dans la recherche d'entreprise :

- La reconnaissance de mots vides propres à une lange qui supprime tous les mots fréquemment utilisés, tels *a* et *the* d'une requête comportant plusieurs mots. Le dictionnaire de mots vides qui existe pour chaque langue ne peut pas être modifié par les utilisateurs. Cette reconnaissance des mots vides est effectuée automatiquement pour toutes les requêtes afin d'améliorer la pertinence de la recherche.
- La reconnaissance des mots vides personnalisée ou définie par l'utilisateur qui supprime du vocabulaire propre à l'entreprise des requêtes. Ce dictionnaire de mots vides défini par l'administrateur peut contenir uniquement du vocabulaire spécial. Le dictionnaire de mots vides défini par l'utilisateur ne remplace pas les dictionnaires de mots vides propres à chaque langue de la recherche d'entreprise qui contiennent des mots communs. Les dictionnaires de mots vides définis par l'utilisateur ne tiennent pas compte de la langue.

Les mots vides définis par l'utilisateur comprennent généralement des termes incluant plusieurs mots, des noms de produit par exemple (tel *OmniFind Enterprise Edition*). Les termes incluant plusieurs mots contenus dans le dictionnaire de mots vides sont correctement identifiés dans les requêtes utilisateur et ne sont pas placés entre guillemets.

Les termes composés des langues germaniques sont également correctement identifiés dans les requêtes. Un terme composé est constitué de deux ou plusieurs mots utilisés comme un seul. Les termes composés lexicalisés, tels *Reisebüro* (agence de voyage) ne sont pas considérés comme étant des termes composés.

Les termes composés d'une requête sont fractionnés en termes individuels. Si un des termes qui compose le terme se trouve dans le dictionnaire de mots vides, le terme composé est alors supprimé de la requête.

Par exemple, le terme de requête `Versicherungspolice` (police d'assurance) renvoie des documents qui contiennent les termes composés *Lebensversicherungspolice* (police d'assurance vie) et *Haftpflichtversicherungspolice* (police d'assurance responsabilité civile). Même si le mot *Police* est répertorié dans le dictionnaire de mots vides, le terme composé de la requête `Versicherungspolice` n'est pas supprimé de la requête.

Vous devez dresser la liste du vocabulaire propre à l'entreprise dans un fichier XML que vous devez ensuite convertir en dictionnaire de mots vides afin qu'il puisse être ajouté au système de recherche d'entreprise.

Vous pouvez sélectionner dans la console d'administration de recherche d'entreprise le dictionnaire de mots vides à utiliser. Vous pouvez sélectionner un dictionnaire de mots vides pour chaque collection. Un dictionnaire de mots vides peut être partagé par plusieurs collections.

Création d'un fichier XML pour les mots vides

Pour retirer du vocabulaire propre à l'entreprise des requêtes, vous devez définir des mots vides dans un fichier XML.

A propos de cette tâche

Le fichier XML qui répertorie les mots vides doit se conformer à un schéma spécifique indiqué dans le document XML. Voici un exemple de fichier XML pour mots vides :

```
<?xml version="1.0" encoding="UTF-8"?>
<stopWords xmlns="http://www.ibm.com/of/83/stopwordbuilder/xml">
   <stopWord>OmniFind Edition</stopWord>
   <stopWord>WAS</stopWord>
   <stopWord>...</stopWord>
</stopWords>
```

Restrictions

Un mot vide peut inclure des espaces mais ne peut pas inclure de caractères de ponctuation, tels une virgule (,) ou une barre verticale (|) car ces caractères peuvent entrer en conflit avec la syntaxe de requête de recherche d'entreprise.

Il n'est pas nécessaire d'énumérer les normalisations du terme, comme la suppression des accents ou des trémas allemands (Umlaut). La recherche d'entreprise gère automatiquement la normalisation. Par exemple, si vous souhaitez inclure le terme météo en tant que mot vide, il n'est pas nécessaire d'inclure également le terme METEO.

Procédure

Pour créer une liste de mots vides pour la recherche d'entreprise, procédez comme suit :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML pouvant valider les éléments XML. Le schéma XSD du fichier XML est appelé `stopWords.xsd` et se trouve dans votre installation de recherche d'entreprise à l'emplacement *RACINE_INSTALL_RECHERCHE_ENTREPRISE*`/packages/uima/configuration_xsd/`.
2. Ajoutez un élément `<stopWord>` pour chaque mot devant être traité comme un mot vide.

   Vérifiez que vos mappages sont inclus dans un élément `<stopWords xmlns="http://www.ibm.com/of/83/stopwordbuilder/xml">`. L'espace de nom (indiqué dans l'attribut xmlns) doit être exactement identique à celui affiché.
3. Répétez la procédure précédente jusqu'à ce que vous ayez spécifié l'ensemble des mots vides à retirer des requêtes lorsque les utilisateurs effectuent des recherches dans des collections.
4. Sauvegardez et quittez le fichier XML.

Une fois que vous avez créé le fichier XML, vous devez le convertir en un dictionnaire de mots vides afin qu'il puisse être ajouté au système de recherche d'entreprise.

Création d'un dictionnaire de mots vides

Une fois que vous avez créé ou mis à jour une liste de mots vides dans un fichier XML, vous devez convertir ce dernier en un dictionnaire de mots vides.

A propos de cette tâche

Pour créer un dictionnaire de mots vides, utilisez l'outil de ligne de commande appelé `esstopworddictbuilder`, fourni avec OmniFind Enterprise Edition. Il se trouve dans le répertoire *ES_INSTALL_ROOT*`/bin`.

L'outil traite un fichier XML qui répertorie les mots vides et génère un dictionnaire de mots vides. Le dictionnaire doit avoir le suffixe `.dic`. Par exemple, `c:\mydictionaries\productstopwords.dic`.

L'emplacement par défaut des fichiers est le répertoire à partir duquel le script est appelé. S'il existe un dictionnaire portant le même nom, le script génère une erreur.

La taille maximale d'un `.dic` dans la recherche d'entreprise est de 8 Mo.

Procédure

Pour créer un dictionnaire de mots vides pour la recherche d'entreprise, procédez comme suit :

1. Sur le serveur d'index, connectez-vous en tant qu'administrateur de recherche d'entreprise. Cet ID utilisateur a été indiqué lors de l'installation de OmniFind Enterprise Edition.
2. Entrez la commande suivante, où *fichier_XML* correspond au chemin complet du fichier XML qui contient la liste des mots vides et *fichier_DIC* au chemin complet du dictionnaire de mots vides.

   AIX, Linux ou Solaris : `esstopworddictbuilder.sh` *fichier_XML fichier_DIC*
   Windows : `esstopworddictbuilder.bat` *fichier_XML fichier_DIC*

Une fois que vous avez créé un dictionnaire de mots vides, utilisez la console d'administration de recherche d'entreprise pour ajouter le dictionnaire au système de recherche d'entreprise et l'associer à une ou à plusieurs collections.

Seul le fichier `.dic` généré est téléchargé vers le système de recherche d'entreprise. Assurez-vous que le fichier XML source est stocké dans un environnement dont l'accès est contrôlé et effectuez une sauvegarde régulière du fichier. Ce fichier XML est requis pour la mise à jour du dictionnaire de mots vides.

Dictionnaires de mots avec degrés de pondération personnalisés

Vous pouvez définir des termes spécifiques ou des termes comportant plusieurs mots qui augmentent ou réduisent la valeur de classement du document dans lequel apparaît le terme.

Chaque terme du dictionnaire de mots avec degré de pondération est associé à un facteur de pondération pouvant aller de -10 à +10. Un facteur de pondération élevé est attribué aux termes que vous souhaitez particulièrement voir dans les documents de résultat. Une valeur faible est attribuée aux termes que vous ne souhaitez pas inclure dans les résultats ou qui sont associés à des termes possédant un degré de pondération plus élevé. Les valeurs -1, 0 et 1 n'ont aucun effet de pondération.

Si un terme de requête répertorié dans le dictionnaire de mots avec degré de pondération avec un facteur de pondération particulier apparaît dans un document extrait, la valeur de classement du document est augmentée ou diminuée en fonction de la valeur de pondération. La valeur de pondération attribuée à un terme est relative car elle peut varier en fonction d'autres facteurs. Si la pondération B1 est attribuée au terme X et que la pondération B2 est attribuée au terme Y et que B1 > B2, alors pondération(X) >= pondération(Y).

Un mot avec degré de pondération comprend généralement des termes incluant plusieurs mots, des noms de produit par exemple (tel *OmniFind Enterprise Edition*). Les termes incluant plusieurs mots contenus dans le dictionnaire de mots avec degré de pondération sont correctement identifiés dans les requêtes utilisateur et ne sont pas placés guillemets.

Les dictionnaires de mots avec degrés de pondération ne tiennent pas compte de la langue.

Les termes composés des langues germaniques sont également correctement identifiés dans les requêtes. Un terme composé est constitué de deux ou plusieurs mots utilisés comme un seul. Les termes composés lexicalisés, tels *Reisebüro* (agence de voyage) ne sont pas considérés comme étant des termes composés.

Les termes composés d'une requête sont fractionnés en termes individuels. Si des valeurs de pondération sont associées aux termes individuels d'un mot composé, les documents extraits sont classés bien que la valeur attribuée soit inférieure à la valeur que le terme aurait s'il apparaissait seul dans le document (et pas dans un mot composé). Ainsi, la portée de la recherche est élargie, ce qui permet de meilleurs résultats lorsque peu de documents contiennent le terme composé complet.

Par exemple, le terme de requête `Versicherungspolice` (police d'assurance) renvoie des documents qui contiennent les termes composés *Lebensversicherungspolice* (police d'assurance vie) et *Haftpflichtversicherungspolice* (police d'assurance responsabilité civile). Si le mot *Police* (police) existe dans le dictionnaire de mots avec degré de pondération, une valeur de pondération est attribuée au document contenant le terme de requête composé `Versicherungspolice`.

Vous devez dresser la liste des termes avec leur valeur de pondération dans un fichier XML que vous pouvez alors convertir en dictionnaire de mots avec degré de pondération, afin qu'il puisse être ajouté au système de recherche d'entreprise.

Vous pouvez avoir recours à la console d'administration de recherche d'entreprise pour sélectionner le dictionnaire de mots avec degré de pondération à utiliser. Vous pouvez sélectionner un dictionnaire de mots avec degré de pondération par collection. Un dictionnaire de mots avec degré de pondération peut être partagé par plusieurs collections.

Création d'un fichier XML pour les mots avec degré de pondération

Pour réduire ou augmenter l'importance de certains documents de résultat, vous devez indiquer les mots qui influencent le classement des documents dans un fichier XML.

A propos de cette tâche

Le fichier XML qui répertorie les mots avec degré de pondération doit se conformer à un schéma spécifique indiqué dans le fichier XML. Voici un exemple de fichier XML pour mots avec degré de pondération :

```
<?xml version="1.0" encoding="UTF-8"?>
<boostTerms xmlns="http://www.ibm.com/of/83/boostbuilder/xml">
    <!-- regrouper les termes avec degré de pondération par valeur de pondération-->
   <boostTermList boost="5">
    <!-- chaque terme peut indiquer l'extension synonyme séparément-->
     <term useVariants="true">OmniFind Edition</term>
     <term useVariants="false">Edition</term>
    <term>OmniFind</term>
   </boostTermList>
   <boostTermList boost="8">
        <term useVariants="true">WAS</term>
        <term>term9</term>
    </boostTermList>
</boostTerms>
```

Restrictions

Vous pouvez regrouper les termes partageant la même valeur de pondération dans un élément `<boostTermList>`, mais une valeur de pondération peut survenir plusieurs fois, par exemple, si vous souhaitez trier les mots avec degré de pondération par ordre alphabétique dans le fichier XML.

Un mot avec degré de pondération peut inclure des espaces mais ne peut pas inclure de caractères de ponctuation, tels une virgule (,) ou une barre verticale (|) car ces caractères peuvent entrer en conflit avec la syntaxe de requête de recherche d'entreprise.

Les termes avec degré de pondération ont des variantes, tels des acronymes ou des abréviations. Vous pouvez énumérer toutes les variantes dans le dictionnaire de mots avec degré de pondération. Toutefois, si vous envisagez d'utiliser un dictionnaire de synonymes ainsi qu'un dictionnaire de mots avec degré de pondération et que vous avez déjà ajouté des termes et leurs variantes au dictionnaire de synonymes, il n'est pas nécessaire d'ajouter ces variantes à la liste des mots avec degré de pondération. A la place, il vous suffit d'attribuer la valeur true à l'attribut `useVariants` pour la variante que vous ajoutez au dictionnaire de mots avec degré de pondération. Toutes les variantes de ce terme répertoriées dans

le dictionnaire de synonymes qui apparaissent dans un des documents extraits influencent la valeur de classement attribuée à ces documents.

Il n'est pas nécessaire d'énumérer les normalisations du terme, comme la suppression des accents ou des trémas allemands (Umlaut). La recherche d'entreprise gère automatiquement la normalisation. Par exemple, si vous souhaitez inclure le terme météo en tant que terme avec degré de pondération, il n'est pas nécessaire d'inclure également le terme METEO.

Procédure

Pour créer une liste de mots avec degré de pondération pour la recherche d'entreprise, procédez comme suit :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix. Le schéma XSD du fichier XML est appelé `boostTerms.xsd` et se trouve dans votre installation de recherche d'entreprise à l'emplacement *RACINE_INSTALL_RECHERCHE_ENTREPRISE*`/packages/uima/configuration_xsd/`.
2. Incluez vos mappages dans un élément `<boostTerms xmlns="http://www.ibm.com/of/83/boostbuilder/xml">`. L'espace de nom (indiqué dans l'attribut xmlns) doit être exactement identique à celui affiché.
3. Ajoutez un élément `<boostTermList>` pour regrouper tous les termes qui partagent la valeur de pondération indiquée.

   Les valeurs de pondération vont de -10 à 10. Par exemple, `<boostTermList boost="-5">` or `<boostTermList boost="5">`.

   L'importance des documents qui contiennent les termes indiqués est augmentée ou réduite en fonction de la valeur de pondération indiquée.
4. Ajoutez un élément `<term>` pour chaque terme qui utilise la valeur de pondération indiquée.

   Si vous souhaitez inclure des variantes d'un mot avec degré de pondération répertoriées dans le dictionnaire de synonymes, affectez la valeur true à l'attribut `useVariants` de l'élément `<term>`. La valeur par défaut est false. Si aucune variante n'est disponible dans le dictionnaire de synonymes, aucun message d'erreur n'est généré.
5. Répétez la procédure précédente jusqu'à ce que vous ayez indiqué l'ensemble des termes utilisés comme mots avec degré de pondération lorsque les utilisateurs effectuent des recherches dans les collections de recherche d'entreprise.
6. Sauvegardez et quittez le fichier XML.

Une fois que vous avez créé le fichier XML, vous devez le convertir en un dictionnaire de mots avec degré de pondération afin qu'il puisse être ajouté au système de recherche d'entreprise.

Création d'un dictionnaire de mots avec degré de pondération

Une fois que vous avez créé ou mis à jour une liste de mots avec degré de pondération dans un fichier XML, vous devez convertir ce dernier en un dictionnaire de mots avec degré de pondération.

A propos de cette tâche

Pour créer un dictionnaire de mots avec degré de pondération, utilisez l'outil de commande appelé `esboostworddictbuilder`, fourni avec OmniFind Enterprise Edition. Il se trouve dans le répertoire `RACINE_INSTALL_RECHERCHE_ENTREPRISE/bin` .

L'outil traite un fichier XML qui répertorie les mots avec degré de pondération et génère un dictionnaire de mots avec degré de pondération. Le dictionnaire doit avoir le suffixe `.dic`. Par exemple, `c:\mydictionaries\productboostwords.dic`.

L'emplacement par défaut des fichiers est le répertoire à partir duquel le script est appelé. S'il existe un dictionnaire portant le même nom, le script génère une erreur.

La taille maximale d'un `.dic` dans la recherche d'entreprise est de 8 Mo.

Procédure

Pour créer un dictionnaire de mots avec degré de pondération pour la recherche d'entreprise, procédez comme suit :

1. Sur le serveur d'index, connectez-vous en tant qu'administrateur de recherche d'entreprise. Cet ID utilisateur a été indiqué lors de l'installation de OmniFind Enterprise Edition.
2. Entrez la commande suivante, où *fichier_XML* correspond au chemin complet du fichier XML qui contient la liste des mots avec degré de pondération et *fichier_DIC*au chemin complet du dictionnaire de mots avec degré de pondération. Si vous souhaitez également utiliser un dictionnaire de synonymes, ajoutez le chemin complet de ce dernier après le nom du dictionnaire de mots avec degré de pondération. L'attribution d'un nom au dictionnaire de synonymes est facultative.

   UNIX : `esboostworddictbuilder.sh`
   fichier_XML fichier_DIC fichier_SYNDIC
   Windows : `esboostworddictbuilder.bat`
   fichier_XML fichier_DIC fichier_SYNDIC

Une fois que vous avez créé un dictionnaire de mots avec degré de pondération, utilisez la console d'administration de recherche d'entreprise pour ajouter le dictionnaire au système de recherche d'entreprise et l'associer à une ou à plusieurs collections.

Seul le fichier `.dic` généré est téléchargé vers le système de recherche d'entreprise. Assurez-vous que le fichier XML source est stocké dans un environnement dont l'accès est contrôlé avec la stratégie de sauvegarde appropriée appliquée. Ce fichier XML est requis pour la mise à jour du dictionnaire de mots avec degré de pondération.

Tâches associées

«Création d'un dictionnaire de synonymes», à la page 67

Analyse de texte incluse dans la recherche d'entreprise

L'analyse de texte de la recherche d'entreprise inclut la détection de la langue du document et la segmentation.

Lorsqu'un document est traité, la recherche d'entreprise détermine la langue du document et fragmente le texte d'entrée en unités distinctes ou marqueurs sémantiques.

Lors d'une recherche, l'utilisateur ou une application doit sélectionner manuellement la langue de la requête. La chaîne de la requête est segmentée, analysée et recherchée dans l'index.

L'analyse de la chaîne de requête et du document peut être fractionnée de la manière suivante :

- Support n'utilisant pas le dictionnaire de base. Inclut une segmentation n-gram et par espaces. La prise en charge n'utilisant pas le dictionnaire de base contient également la segmentation par phrases.
- Support linguistique utilisant des dictionnaires. Inclut une segmentation par mots et par phrases et la lemmatisation.

  Le traitement linguistique implique l'analyse lexicale, processus de création de représentations alternatives du texte d'entrée qui associe toutes les données des dictionnaires disponibles aux marqueurs sémantiques reconnus dans le texte d'entrée. Vous pouvez améliorer la pertinence de la recherche en utilisant un traitement avancé de la langue.

  Concepts associés

  «Identification de la langue»

  «Support linguistique pour la segmentation effectuée sans dictionnaire», à la page 78

Identification de la langue

Avant que la segmentation par mot et par phrase, la normalisation des caractères ou la lemmatisation puissent avoir lieu, la recherche d'entreprise doit déterminer la langue du document source.

La recherche d'entreprise peut automatiquement détecter les langues suivantes :

Tableau 10. Langues prises en charge pour l'identification automatique de la langue

Afrikaans	Arabe	Balinais
Basque	Catalan	Chinois (traditionnel et simplifié)
Tchèque	Danois	Néerlandais
Anglais	Finnois	Français
Allemand	Grec	Hébreu
Islandais	Irlandais (gaélique)	Italien
Japonais	Coréen	Malais
Norvégien (Bokmål)	Polonais	Portugais

Tableau 10. Langues prises en charge pour l'identification automatique de la langue (suite)

Roumain	Russe	Espagnol
Suédois	Tagalog	Thaï
Turc	Vietnamien	

Le processus linguistique de la recherche d'entreprise détecte la langue d'un document source lors de l'indexation et non lors du traitement de la requête.

Dans la recherche d'entreprise, vous pouvez choisir l'option de détection automatique de la langue ou sélectionner une langue à utiliser.

Si vous sélectionnez la détection automatique de la langue et que l'analyseur syntaxique ne peut pas la déterminer, il utilise la langue indiquée lors de la création du moteur de balayage dans la console d'administration de recherche d'entreprise.

Si vous ne sélectionnez pas la détection automatique de la langue, la langue que vous indiquez est toujours utilisée. Vous indiquez la langue du document en modifiant les propriétés du moteur de balayage dans la console d'administration de la recherche d'entreprise. La langue par défaut est l'anglais.

Les documents pour lesquels il n'existe aucun dictionnaire spécifique à une langue seront traités à l'aide d'une technologie linguistique de base, telle que la segmentation à l'aide d'espaces ou la segmentation n-gram.

La technologie de détection de langue de recherche d'entreprise est la plus adaptée pour les documents en une seule langue. Si un document est rédigé en plusieurs langues, une tentative de détection de la langue dominante du document est effectuée. Toutefois, les résultats de l'analyse ne sont pas toujours satisfaisants.

La langue d'un document peut être utilisée afin de restreindre les résultats de la recherche aux documents rédigés dans une langue spécifique. Par exemple, si vous recherchez des documents sur Jacques Chirac dans une collection de documents multilingues, vous pouvez limiter les résultats de la recherche aux seuls documents écrits en français. La définition de la langue de vos documents de sortie est une option de recherche avancée que vous pouvez sélectionner sur la console d'administration de la recherche d'entreprise.

Concepts associés

«Analyse de texte incluse dans la recherche d'entreprise», à la page 77

«Support linguistique pour la segmentation effectuée sans dictionnaire»

Support linguistique pour la segmentation effectuée sans dictionnaire

Pour les documents rédigés dans des langues qui ne sont pas prises en charge par la technologie d'analyse lexicale, la recherche d'entreprise fournit un support de base sous la forme de segmentation n-gram et d'espace de type Unicode.

segmentation d'espace de type Unicode
: Cette méthode de traitement linguistique utilise les espaces entre les mots comme délimiteurs.

Segmentation n-gram

Cette méthode de traitement linguistique traite les séquences de *n* caractères comme un seul mot. Cette méthode simple de segmentation est suffisante pour la plupart des tâches d'extraction.

Ces méthodes n'utilisent aucun dictionnaire et n'incluent aucune technologie de traitement linguistique sophistiquée, telles la réduction à la forme de base.

La segmentation N-gram est utilisée pour des langues, telles le thaïlandais, qui n'ont pas recours aux espaces comme délimiteurs. La même méthode s'applique à l'hébreu et à l'arabe. Bien que ces deux langues utilisent des espaces comme délimiteurs, la segmentation n-gram renvoie de meilleurs résultats que ne le fait la forme de base de la segmentation à l'aide d'espaces de type Unicode.

Lorsque vous créez votre collection, vous pouvez également choisir éventuellement de marquer sémantiquement des documents chinois et japonais à l'aide de la segmentation n-gram.

Pour supprimer tout espace, par exemple pour une nouvelle ligne ou une tabulation, durant la segmentation n-gram, vous devez activer les paramètres du fichier collection.properties dans *ES_NODE_ROOT*/master_config/<CollectionID>.parserdriver avant de commencer à analyser les documents. Les paramètres requis pour supprimer les espaces incluent :

- **removeCjNewLineChars** : S'il est défini sur *true*, ce paramètre supprime les séquences de caractères de nouvelle ligne ou de tabulation apparaissant entre les caractères chinois ou japonais. La valeur par défaut est `removeCjNewlineChars=false`.
- **removeCjNewLineCharsMode** : S'il est défini sur *all*, ce paramètre supprime les espaces quel que soit le contexte de caractères. Par exemple, les espaces sont aussi supprimés dans le texte anglais. Si vous voulez utiliser cette option, vous devez ajouter le paramètre au fichier de propriétés. Seule la valeur `removeCjNewlineCharsMode=all` est valide, toutes les autres sont ignorées.

Concepts associés

«Analyse de texte incluse dans la recherche d'entreprise», à la page 77

«Identification de la langue», à la page 77

Marquage des caractères numériques comme marqueurs sémantiques n-gram

Pour marquer sémantiquement des caractères numériques, en plus des caractères à deux octets, en tant que marqueurs n-gram, vous devez modifier un paramètre dans le fichier descripteur de l'annotateur.

A propos de cette tâche

La gestion par défaut des caractères numériques du marqueur sémantique n-gram et des espaces consiste à traiter tous les caractères numériques comme des marqueurs sémantiques segmentés par des espaces. Pour marquer sémantiquement des caractères numériques comme des marqueurs sémantiques n-gram, vous devez modifier le paramètre du mode n-gram du fichier descripteur de l'annotateur. Vous ne pouvez pas modifier ce paramètre en utilisant la console d'administration de la recherche d'entreprise.

Conseil : Il existe trois modes de marquage sémantique n-gram : normal, numérique et complet. Cette procédure explique comment activer le marquage

n-gram numérique. Pour plus d'informations sur la configuration de la prise en charge du marquage sémantique n-gram complet dans des collections de recherche d'entreprise et sur la gestion des caractères dans les collections configurées pour une prise en charge n-gram complète, consultez la page http://www.ibm.com/support/docview.wss?rs=63&uid=swg27011088.

Procédure

Le paramètre du mode n-gram par défaut est appelé normal et traite les caractères numériques et SBCS comme des caractères segmentés par des espaces. Pour activer le mode n-gram numérique, procédez comme suit :

1. Arrêtez l'analyseur syntaxique pour votre collection.
2. Arrêtez l'exécution pour votre collection.
3. Ouvrez le fichier descripteur d'annotateur appelé jtok.xml dans le répertoire *RACINE_NOEUD_ES*/master_config/*ID_collecte*.parserdriver/specifiers, où *ID_collecte* est l'ID spécifié pour la collection (ou attribué par le système) lors de sa création.
4. Modifiez la définition du paramètre **NgramMode** de normal à numérique.
5. Redémarrez l'analyseur syntaxique pour votre collection.
6. Redémarrez l'exécution.

Support linguistique pour la segmentation effectuée à l'aide d'un dictionnaire

Si la langue d'un document est correctement détectée et que des dictionnaires correspondant à cette langue sont disponibles, alors le traitement linguistique approprié est appliqué.

Le processus permettant de séparer le texte d'entrée en des unités lexicales distinctes est appelé segmentation. Ce processus inclut certaines des activités de traitement linguistique suivantes :

Segmentation de mots
La segmentation de mots est employée pour les langues qui n'utilisent pas d'espace (ou délimiteur) entre les mots, pour le japonais ou le chinois, par exemple.

Lemmatisation
La lemmatisation est une forme de traitement linguistique qui détermine le lemme de chaque mot du texte. Le *lemme* d'un mot inclut sa forme de base ainsi que les versions déclinées qui partagent la même classe de mots. Par exemple, le lemme de *go* inclut *go*, *goes*, *went*, *gone* et *going*. Les lemmes de noms comportent le singulier et le pluriel (*calf* et *calves*, par exemple). Les lemmes d'adjectifs incluent les formes comparatives et superlatives (*good*, *better* et *best*, par exemple). Les lemmes des pronoms rassemblent différentes formes du même pronom (*I*, *me*, *my* et *mine*, par exemple).

La lemmatisation nécessite une dictionnaire pour l'indexation et pour la recherche.

La recherche d'entreprise indexe les lemmes et les mots décline et effectue une lemmatisation de tous les mots déclinés dans une requête. La lemmatisation améliore la qualité de la recherche en recherchant des documents qui contiennent des variantes d'un terme décliné dans la requête. Par exemple, les documents contenant le mot *mice* sont trouvés lorsque la requête inclut le mot *mouse*.

Fragmentation des contractions

La qualité de la recherche est améliorée en identifiant les contractions et en les fractionnant. Par exemple :

wouldn't est fractionné en *would + not*
Horse's est fractionné en *Horse + 's*

Identification clitique

Les clitiques constituent une forme spéciale de contractions et la qualité de la recherche est améliorée en déterminant les différentes parties de composant. Un *clitique* est un élément qui se comporte comme un affixe et comme un mot. Toutefois, il est difficile d'identifier les clitiques car ils font également partie de la formation du mot. Contrairement à d'autres phénomènes morphologiques (structure des mots), les clitiques se trouvent dans une structure syntaxique et leur lien aux mots ne fait pas partie des règles de formation de mots. Par exemple :

reparti-lo-emos est constitué des composants *repartir + lo + emos*
l'avenue est constitué des composants *le + avenue*
dell'arte est constitué des composants *dello + arte*.

Reconnaissance des caractères non alphabétiques

Les processus linguistiques reconnaissent les caractères non alphabétiques. Selon la logique interne dépendant de la langue, certains caractères non alphabétiques sont renvoyés en tant qu'unités lexicales de types différents et d'autres sont regroupées.

Par exemple, les apostrophes sont considérées comme parties intégrantes d'un mot dans le cas de clitiques mais comme des points dans le cas d'abréviations inconnues. Les URL, adresses électroniques et dates sont divisées en plusieurs jetons.

Reconnaissance des abréviations

Les processus linguistiques reconnaissent les abréviations qui se trouvent dans le dictionnaire en tant qu'une seule unité lexicale. Si l'abréviation ne se trouve pas dans le dictionnaire, l'abréviation est reconnus comme élément lexical mais aucune information de dictionnaire n'est associé à l'abréviation.

Une reconnaissance correcte des abréviations est primordiale pour la reconnaissance des phrases. Par exemple, le point placé à la fin d'une abréviation ne représente pas forcément la fin d'une phrase.

Reconnaissance du marqueur de fin de phrase

Les processus linguistiques identifient correctement les marqueurs de fin de phrase pour la segmentation des phrases.

Le support linguistique à l'aide de dictionnaires est disponible pour les langues suivantes :

Tableau 11. Langues prises en charge

Arabe	Italien
Chinois (simplifié et traditionnel)	Japonais
Tchèque	Coréen
Danois	Norvégien (Bokmål)
Néerlandais	Polonais
Anglais	Portugais (national et brésilien)

Tableau 11. Langues prises en charge (suite)

Finnois	Russe
Français (national et canadien)	Espagnol
Allemand (national et suisse)	Suédois
Grec	

Concepts associés

«Segmentation des mots en japonais»

«Variantes orthographiques en japonais»

Segmentation des mots en japonais

S'il a été détecté que le document ou la chaîne de requête est en japonais, la recherche d'entreprise effectue une segmentation des mots appropriée en utilisant la technologie d'analyse morphologique optimisée pour le japonais.

Cette optimisation est par exemple illustrée par la décomposition des mots. La langue japonaise utilise un grand nombre de mots composés. Ces mots sont décomposés en marques sémantiques de taille optimale afin d'obtenir de meilleurs résultats de recherche. Les mots déclinés et les prépositions sont également décomposés afin d'améliorer les performances de la recherche.

Concepts associés

«Support linguistique pour la segmentation effectuée à l'aide d'un dictionnaire» , à la page 80

«Variantes orthographiques en japonais»

Variantes orthographiques en japonais

La langue japonaise utilise un grand nombre de variantes orthographiques. Les variantes Katakana sont les plus importantes car Katakana est souvent utilisé pour orthographier et prononcer des mots étrangers. Un grand nombre de variantes Katakana sont souvent utilisées en japonais.

La recherche d'entreprise utilise un dictionnaire de variantes pour mapper des variantes Katakana classiques à leurs formes de base afin que tous les documents, y compris ceux avec des variantes orthographiques du caractère Katakana dans la chaîne de requête, soient trouvés.

La recherche d'entreprise prend également en charge les variantes Okurigana classiques, fins de mots Kanji écrits en Hiragana.

Concepts associés

«Support linguistique pour la segmentation effectuée à l'aide d'un dictionnaire» , à la page 80

«Segmentation des mots en japonais», à la page 82

Suppression des mots vides

Dans la recherche d'entreprise, tous les mots vides, par exemple des mots communs, *un* et *le* par exemple, sont supprimés des requêtes comportant plusieurs mots afin d'améliorer les performances de la recherche.

La reconnaissance des mots vides pour le japonais s'effectue à l'aide d'informations grammaticales, par exemple, la recherche d'entreprise reconnaît si le mot est un nom ou un verbe. Pour les autres langues, la recherche d'entreprise utilise des listes spéciales.

Aucun mot vide n'est supprimé durant le traitement de la requête si :

- Tous les mots d'une requête sont des mots vides. Si tous les termes de la requête sont supprimés lors du traitement des mots vides, alors l'ensemble de résultats est vide. Pour garantir que les résultats de la recherche sont renvoyés, la suppression de mots vides est désactivée lorsque tous les mots de la requête sont des mots vides. Par exemple, si le mot *car* est un mot vide et que vous recherchez *car*, alors les résultats de la recherche contiennent les documents correspondant au mot *car*. Si vous recherchez *car buick*, les résultats de la recherche contiennent uniquement les documents correspondant au mot *buick*.
- Le mot d'une requête est précédé du signe plus (+).
- Le mot fait partie d'une correspondance exacte.
- Le mot est dans une phrase, par exemple, ″I love my car″.

Concepts associés

«Normalisation des caractères»

Normalisation des caractères

La normalisation des caractères est un processus qui peut améliorer le rappel. En améliorant le rappel à l'aide de la normalisation des caractères, un plus grand nombre de documents est extrait même si les documents ne correspondent pas exactement à la requête.

La recherche d'entreprise utilise la normalisation de compatibilité Unicode qui inclut la normalisation des caractères asiatiques demi-largeur à largeur standard.

La recherche d'entreprise supprime également les points qui sont utilisés en tant que délimiteurs de mots composés en japonais.

D'autres formes de normalisation de caractères incluent :

Normalisation majuscules/minuscules
Par exemple, la recherche de documents comportant *USA* lorsque la chaîne *usa* a été indiquée.

Développement des trémas allemands (Umlaut)
Exemple : Recherche des documents qui contiennent *schoen* lors de la recherche de *schön*.

Suppression des accents
Exemple : Recherche des documents contenant le caractère *é* lors de la recherche du caractère *e*.

Suppression d'autres signes diacritiques
Exemple : Recherche de documents contenant le caractère *ç* lors de la recherche du caractère *c*.

Développement de la ligature
Exemple : Recherche des documents contenant les caractères *Æ* lors de la recherche des caractères *ae*.

Toutes les normalisations fonctionnent dans les deux sens. Vous pouvez trouver des documents contenant *usa* lorsque vous recherchez *USA*, des documents contenant des mots avec le caractère *e* lorsque vous recherchez le caractère *é*, etc. Ces normalisations peuvent être associées les unes aux autres. Par exemple, vous pouvez trouver des documents comportant le terme *météo* lorsque vous recherchez *METEO*.

Les normalisations sont effectuées en fonction des propriétés des caractères Unicode et ne dépendent pas des langues. Par exemple, la recherche d'entreprise prend en charge la suppression des caractères diacritiques pour l'hébreu et le développement de la ligature pour l'arabe.

Concepts associés

«Suppression des mots vides», à la page 83

Annotateur d'expressions régulières

L'annotateur d'expressions régulières vous permet d'effectuer une analyse de texte personnalisée sans avoir à implémenter votre propre moteur d'analyse de texte. En fonction d'un jeu de règles (expressions régulières) que vous pouvez définir vous-même, l'annotateur d'expressions régulières détecte les structures d'informations des documents texte et crée des annotations sur les informations détectées dans la structure d'analyse commune.

L'annotateur d'expressions régulières détecte les entités ou unités d'informations des documents texte, par exemple, les numéros de téléphone, les codes de produit, les numéros de bâtiment ou de chambre ou les adresses, en fonction des expressions régulières. Si l'une des expressions régulières correspond à des parties du texte du document, l'annotateur d'expressions régulières crée les annotations correspondantes qui couvrent la partie correspondante des informations. Ces annotations sont stockées dans la structure d'analyse commune et peuvent ensuite être recherchées en mappant ces résultats d'analyse à l'index de la recherche d'entreprise, à l'aide d'un fichier de mappage de la structure d'analyse commune à l'index. Sinon, un fichier de mappage de la structure d'analyse commune à la base de données peut être créé pour stocker les annotations dans une base de données compatible JDBC.

Le jeu de règles (expressions régulières) que vous définissez est stocké dans un fichier de configuration XML (également appelé le fichier du jeu de règles). L'annotateur d'expressions régulières contient la logique d'analyse qui traite ces expressions régulières. Il prend en charge la syntaxe d'expression régulière de Java 1.4.

La description du système de types de l'annotateur d'expressions régulières doit définir les types d'annotation et fonctions utilisés et créés par l'annotateur d'expressions régulières. Selon la complexité de la zone d'application de l'annotateur d'expressions régulières (par exemple, si davantage de types qu'il n'en est défini dans l'annotateur d'expressions régulières fourni sont requis), des fonctions supplémentaires d'entrée et sortie doivent être définies dans le descripteur de l'annotateur d'expressions régulières. Les types utilisés dans le descripteur doivent correspondre aux types de la description du système de types de l'annotateur.

L'annotateur d'expressions régulières est inclus dans la recherche d'entreprise en tant que fichier PEAR (Processing Engine ARchive) déployable, configuré avec des exemples de règles pour détecter les numéros de téléphone, les URL et les adresses électroniques.

Concepts associés

«Le fichier du jeu de règles», à la page 88

Tâches associées

«Définition des règles d'expression régulière», à la page 89

Référence associée

«Le descripteur d'annotateur», à la page 93

«Journalisation», à la page 97

Recherche sémantique facilitée à l'aide de l'annotateur d'expressions régulières

La recherche d'entreprise inclut le moteur d'analyse d'expressions régulières préconfiguré avec un jeu de règles permettant de détecter des numéros de téléphone, des URL et des adresses électroniques dans des documents texte.

Vous pouvez utiliser cet exemple de configuration du moteur d'analyse d'expressions régulières pour permettre à la recherche d'entreprise de trouver des numéros de téléphone dans les documents sans rechercher le mot clé *numéro de téléphone* dans les documents. Pour lancer une requête sur les constructions détectées par l'annotateur d'expressions régulières, un exemple de fichier de mappage de la structure d'analyse commune à l'index est également fourni. De plus, une méthode simple est expliquée, pour vous permettre de générer de puissantes requêtes sémantique à travers de simples mots clés. Cette méthode utilise la prise en charge des synonymes de la recherche d'entreprise pour étendre automatiquement les requêtes par mot clé simple en requêtes sémantiques. Un exemple de dictionnaire des synonymes, illustrant ce mécanisme, est fourni. Vous pouvez trouver tous les fichiers dont vous avez besoin pour utiliser l'annotateur d'expressions régulières avec l'exemple de configuration dans `ES_INSTALL_ROOT/packages/uima/regex`.

Pour de nombreux scénarios d'application, il peut être suffisant de modifier seulement légèrement les règles d'expressions régulières fournies avec l'exemple de configuration afin d'adapter l'annotateur d'expressions régulières à vos besoins.

Toutefois, pour personnaliser totalement l'annotateur, nous vous recommandons d'utiliser le SDK UIMA. Aussi, l'annotateur d'expressions régulières est également inclus dans le module d'annotateurs de base de la recherche d'entreprise, situé dans `ES_INSTALL_ROOT/packages/uima/`.

Tâches associées

«Activation de la recherche sémantique facilitée à l'aide de l'annotateur d'expressions régulières»

«Personnalisation de l'annotateur d'expressions régulières», à la page 92

«Affichage des résultats de l'annotateur de base et de l'analyse de texte personnalisée», à la page 13

Activation de la recherche sémantique facilitée à l'aide de l'annotateur d'expressions régulières

Pour permettre la recherche sémantique facilitée à l'aide des synonymes, vous devez ajouter l'annotateur d'expressions régulières, le fichier de mappage de la structure d'analyse commune à l'index et l'exemple de dictionnaire des synonymes à votre système de recherche d'entreprise et associer ces ressources à votre collection.

Ensuite, l'annotateur d'expressions régulières traitera vos documents durant la phase d'analyse syntaxique, l'indexeur ajoutera les résultats de l'analyse personnalisée à l'index et le service de recherche pourra utiliser le dictionnaire de synonymes sémantique fourni pour rechercher des résultats d'analyse personnalisés à travers des mots clés simples automatiquement étendus en requêtes sémantiques.

Procédure

Pour activer la recherche sémantique facilitée, procédez comme suit :

1. Ajoutez le moteur d'analyse de texte personnalisé d'expressions régulières appelé `of_regex.pear` se trouvant dans *ES_INSTALL_ROOT*`/packages/uima/regex` au système de recherche d'entreprise à l'aide de la console d'administration de la recherche d'entreprise.
2. Associez le moteur d'analyse de texte d'expressions régulières à votre collection.
3. Ajoutez le fichier de mappage de la structure d'analyse commune à l'index appelé `of_sample_regex_cas2index.xml` au répertoire *ES_INSTALL_ROOT*`/packages/uima/regex`. Cette action mappe les résultats de l'analyse personnalisée (annotations) que l'annotateur d'expressions régulières produit aux étendues interrogeables de l'index de la recherche d'entreprise. Ensuite, vous pouvez utiliser le fragment XML ou les requêtes XPath pour rechercher ces étendues.
4. Explorez, analysez et indexez votre collection. A ce stade, une fois l'indexation terminée, vous pourriez entrer une requête de recherche XML à l'aide d'une expression de fragment XML, par exemple `@xmlf2::'<#phonenumber>'`, à l'aide de l'application de recherche. Toutefois, l'objectif de l'activation de la recherche sémantique par synonymes est de vous permettre d'utiliser les requêtes telles que `Barbara phone number` et que le système traduise la requête en Barbara @xmlf2::'<#phonenumber>'.
5. Ajoutez l'exemple de dictionnaire de synonymes binaire fourni, appelé `of_sample_synonym_dic.dic` et se trouvant dans le répertoire *ES_INSTALL_ROOT*`/packages/uima/regex`, au système de recherche d'entreprise à l'aide de la console d'administration. Vous pouvez apporter des modifications à l'exemple de dictionnaire XML source ou l'utiliser comme base pour créer votre propre dictionnaire et le convertir en un nouveau fichier de dictionnaire à l'aide de l'outil essyndictbuilder. L'exemple de dictionnaire de synonymes XML s'appelle `of_sample_synonym_dic.xml` et se trouve également dans *ES_INSTALL_ROOT*`/packages/uima/regex`.
6. Associez le dictionnaire de synonymes à votre collection et démarrez (ou redémarrez) le service de recherche pour votre collection.
7. Dans l'application de recherche, sélectionnez l'option pour rechercher automatiquement des synonymes en utilisant l'extension sémantique. Une fois cette option activée, l'application de recherche réécrit vos requêtes de mots clés de base en requêtes de fragments XML et inclut les expressions trouvant les étendues interrogeables qui identifient les numéros de téléphone, les adresses électroniques et les URL.
8. Dans l'application de recherche, entrez une requête demandant un numéro de téléphone, par exemple, `barbara telephone number`. La requête recherche les documents contenant les trois mots clés *barbara*, *telephone* et *number*, ainsi que les documents contenant le mot clé *barbara* et les étendues des nombres et caractères dans les documents qui correspondent aux expressions régulières définies pour un numéro de téléphone. Les mots clés et numéros de téléphone trouvés sont mis en évidence dans les résultats de la recherche.

Vous pouvez voir quels mots clés sont traduits en requêtes sémantiques dans l'exemple de dictionnaire de synonymes fourni.

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">
  <synonymgroup>
   <synonym>telephone number</synonym>
   <synonym>phone number</synonym>
   <synonym>telephone nbr</synonym>
   <synonym>phone nbr</synonym>
```

```
   <synonym>@xmlf2::'&lt;#phonenumber/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
   <synonym>facsimile number</synonym>
   <synonym>fax number</synonym>
   <synonym>facsimile nbr</synonym>
   <synonym>fax nbr</synonym>
   <synonym>@xmlf2::'&lt;#phonenumber/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
   <synonym>e-mail address</synonym>
   <synonym>email address</synonym>
    <synonym>@xmlf2::'&lt;#email/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
   <synonym>URL</synonym>
   <synonym>unified resource locator</synonym>
   <synonym>Web address</synonym>
   <synonym>@xmlf2::'&lt;#url/&gt;'</synonym>
  </synonymgroup>
</synonymgroups>
```

Concepts associés

«Recherche sémantique facilitée à l'aide de l'annotateur d'expressions régulières», à la page 86

Le fichier du jeu de règles

Dans l'annotateur d'expressions régulières, le fichier du jeu de règles XML définit les règles, sous forme d'expressions régulières, utilisées pour analyser le document texte.

Les règles indiquent, dans l'ordre séquentiel, à quel endroit du document l'annotateur doit rechercher quel élément et quelle action il doit prendre si une correspondance est trouvée.

Lorsque l'annotateur d'expressions régulières est appelé, le fichier du jeu de règles XML contenant les masques d'expressions régulières est compilé et comparé aux parties du document texte. Si une correspondance ou correspondance partielle est trouvée, l'annotation associée à la règle spécifique est créée et stockée dans la structure d'analyse commune.

Les types utilisés dans les règles doivent être définis dans la description du système de types de l'annotateur d'expressions régulières.

L'annotateur d'expressions régulières traite une règle à la fois, en commençant par la première règle du fichier du jeu de règles XML. Pour chaque règle, l'expression régulière compilée correspondante est comparée aux annotations créées dans une étape antérieure, par exemple, aux annotations créées par les annotateurs qui ont traité le document avant l'annotateur d'expressions régulières. Les annotations qui correspondent aux règles doivent être du même type que les types de fonctions d'entrée spécifiés dans le descripteur de l'annotateur d'expressions régulières.

Si une correspondance est trouvée, le type d'annotation créé dans la règle qui se déclenche doit également être spécifié en tant que type de fonction de sortie valide dans le descripteur de l'annotateur d'expressions régulières. Les nouvelles annotations créées par une règle antérieure peuvent être utilisées en tant qu'annotations d'entrée pour les règles qui se déclenchent ultérieurement dans le jeu de règles XML.

Concepts associés

«Annotateur d'expressions régulières», à la page 85

Tâches associées

«Définition des règles d'expression régulière»

Référence associée

«Le descripteur d'annotateur», à la page 93

«Journalisation», à la page 97

Définition des règles d'expression régulière

Le jeu de règles définit les expressions régulières comparées au texte du document et les actions que l'annotateur d'expressions régulières doit prendre si un critère correspond.

A propos de cette tâche

Le fichier du jeu de de règles XML doit suivre la syntaxe des règles mise en évidence dans l'exemple suivant. Il s'agit du fichier du jeu de règles pour l'exemple d'annotateur d'expressions régulières qui reconnaît les numéros de téléphone, les URL et les adresses électroniques.

L'élément de niveau supérieur est un élément `<ruleSet>` qui comprend un ou plusieurs éléments `<rule>`. Chaque élément `<rule>` à son tour définit une expression régulière Java comprenant un attribut `regEx` ainsi que les attributs `matchStrategy` et `matchType`. L'action est définie dans l'élément `<createAnnotation>` qui indique l'ID d'annotation et le type d'annotation.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ruleSet.xsd">
 <!-- Numéro de téléphone -->
 <!-- Cette règle correspond à plusieurs formats de numéros de téléphone possibles,
   par exemple, 01234-12345, 01234 / 122-32, (001234)12345,
   +49 (0) 123412345, (123) 123 1234,
   1-800-IBM-4YOU -->
 <rule regEx="(?x)(\s|\b)(
0{1,2}[1-9]{1}[0-9]{1,5}\x20?[-/\\]\x20?[1-9]{1}([0-9]{1,8}-?)
  {1,3}[0-9]{1,}
|\(0[1-9]{1}[0-9]{1,3}\)\x20?[1-9]{1}[0-9]{2,8}
|\(00[1-9]{1}[0-9]{1,8}\)\x20?[1-9]{1}[0-9]{2,10}
|\((0\x20?[1-9]{1}[0-9]{1,3}|00\x20?[1-9]{1}[0-9]{1,8})\)\x20?[1-9]
   {1}[0-9] {1,3}(\x20[0-9]{2,4}){1,5}
|(0\x20?[1-9]{1}[0-9]{1,3}|00\x20?[1-9]{1}[0-9]{1,8})\x20?[/\\]\x20?
   [1-9]{1} [0-9]{1,3}(\x20[0-9]{2,4}){1,5}
|\(?\+[1-9]{1}[0-9]{0,3}\)?([-\x20]|\x20?\(0\))[-\x20]?[1-9]
   {1}[0-9]{1,10}
|\(?\+[1-9]{1}[0-9]{0,3}\)?([-\x20]|\x20?\(0\))[-\x20]?[1-9]
   {1}[0-9]{1,3}[-\x20]([0-9]{2,5}[-\x20]?){1,4}
|(1-)?[0-9]{3}-[0-9]{3}-[0-9]{4}
|\([1-9]{1}[0-9]{2}\)\x20[0-9]{3}[-\x20][0-9]{4}
|1-(800|888|877|866)-([A-Z0-9]{7}|[A-Z0-9]{3}-[A-Z0-9]
   {4}|[A-Z0-9]{4}-[A-Z0-9]{3})
)(?!(\d|\x20\d|-\d))(\s|\b)"
  matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
  <createAnnotation id="phonenumber" type="com.ibm.es.uima.PhoneNumber">
   <begin group="0"/>
   <end group="0"/>
  </createAnnotation>
 </rule>
 <!-- Numéro de téléphone éventuel -->
 <!-- Cette règle correspond aux numéros qui ressemblent à des numéros de téléphone,
        mais qui peuvent être tout autre chose. Par exemple, 0123 1234 123,
```

```
+123456789, 123 123 1234 -->
<rule regEx="(?x)(\s|\b)(
 0[1-9]{1}[0-9]{1,3}\x20[1-9]{1}[0-9]*\x20?([0-9]{2,}\x20?)+
 |00\x20?[1-9]{1}[0-9]{0,3}\x20[1-9]{1}[0-9]{1,3}\x20?[1-9]
  {1}([0-9]{2,}\x20?)+
 |\+[1-9]{1}[0-9]{0,3}[1-9]{1}[0-9]{6,}
 |[1-9]{1}[0-9]{2}\x20[0-9]{3}\x20[0-9]{4}
 )(?!(\d|\x20\d|-\d))(\s|\b)"
 matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
 <createAnnotation id="potential_phonenumber"
    type="com.ibm.es.uima.PotentialPhoneNumber">
  <begin group="0"/>
  <end group="0"/>
 </createAnnotation>
</rule>
<!-- Annotation d'URL -->
<!-- Cette règle correspond aux URL, par exemple, http://www.ibm.com -->
<rule regEx="(?x)(\s|\b)(
   http://[\w\-]+([\.][\w\-]+)+([/][\w\~\(\)\-\?=%\u0026\#]*)*
   |www.[\w\-]+([\.][\w\-]+)+([/][\w\~\(\)\-\?=%\u0026\#]*)*
  )(\s|\b)"
 matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
 <createAnnotation id="url" type="com.ibm.es.uima.URL">
  <begin group="0"/>
  <end group="0"/>
 </createAnnotation>
</rule>
<!-- Annotation d'adresse électronique -->
<!-- Cette règle correspond aux adresses électroniques,
       par exemple, votrenom@domaine.com -->
<rule regEx="(?x)(\s|\b)(
   [a-zA-Z0-9][\w\.-]*[a-zA-Z0-9]@[a-zA-Z0-9]([\.-]?\w)*\.[a-zA-Z]
   {2,3})(\s|\b)"
 matchStrategy="matchAll" matchType="uima.tcas.DocumentAnnotation">
 <createAnnotation id="email" type="com.ibm.es.uima.Email">
  <begin group="0"/>
  <end group="0"/>
 </createAnnotation>
</rule>
</ruleSet>
```

Procédure

Pour créer le jeu de règles XML pour l'annotateur d'expressions régulières qui définit vos expressions régulières personnalisées, procédez comme suit :

1. Créez un fichier XML. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix. Le schéma XSD pour le fichier du jeu de règles XML s'appelle `ruleSet.xsd`, que vous pouvez trouver dans l'installation de la recherche d'entreprise du répertoire *ES_INSTALL_ROOT*`/packages/uima/regex/`.
2. Incluez vos mappages dans un élément `<ruleSet xmlns="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="ruleSet.xsd">`. L'espace de nom est indiqué dans l'attribut xmlns et doit être exactement identique à celui affiché.
3. Ajoutez un élément `<rule>` qui contient un attribut `regEx` contenant le masque d'expression régulière, un attribut`matchStrategy` et un attribut `matchType`.

   L'annotateur prend totalement en charge la syntaxe d'expressions régulières Java 1.4. Pour une introduction aux expressions régulières et pour afficher la syntaxe complète, reportez-vous à la documentation Java à l'adresse http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.

matchStrategy indique comment rechercher, par exemple, si toutes les correspondances doivent être trouvées dans le document ou si la correspondance textuelle doit être une correspondance exacte. Trois stratégies de correspondance différentes sont disponibles :

- matchFirst s'arrête à la première séquence de texte qui correspond au masque normal
- matchAll trouve toutes les séquences de texte dans un document, qui correspondent au masque normal
- matchComplete trouve uniquement les séquences de texte qui correspondent exactement. Par exemple, avec un masque "foo", seul le terme ″foo″ serait mis en correspondance, ″foobar″ ne serait pas considéré comme une correspondance.

matchType détermine le type d'annotation auquel la règle est comparée. Ainsi, vous pouvez restreindre votre expression régulière pour qu'elle corresponde, par exemple, au sein d'une annotation de marqueur sémantique existante. Cela évite de faire correspondre trop de contenu dans une règle. Les types possibles sont les types d'annotations d'entrées autorisés sur l'annotateur (définis dans le descripteur d'annotateur), tels que uima.tt.DocumentAnnotation, uima.tt.ParagraphAnnotation et les types définis par l'utilisateur tels que foo.bar.MyAnnotation. Parfois, le type de sortie d'une règle est utilisé en tant que type d'entrée d'une règle ultérieure. matchType vous permet de restreindre la portée de recherche de certaines règles.

4. Ajoutez un élément <createAnnotation> qui définit l'action que l'annotateur d'expression régulière doit prendre si une correspondance est trouvée.

   Chaque élément createAnnotation présente deux attributs :

   - id identifie l'annotation de façon unique et est utilisé pour référencer l'annotation
   - type indique le type d'annotation créé

5. Ajoutez les éléments de composant suivants, qui définissent la position de correspondance pour l'élément <createAnnotation> :

   - Obligatoire : <begin> indique où la correspondance commence. Cet élément présente deux attributs :
     - Obligatoire : group identifie le groupe de capture Java. Une valeur comprise entre 0 (correspondance de séquence de texte totale) et 9 (plusieurs groupes de capture) peut lui être attribuée
     - Facultatif : location indique une position dans le groupe de correspondance (par rapport au positionnement des parenthèses), start (parenthèse gauche) ou end (parenthèse droite).
   - Obligatoire : <end> indique où la correspondance se termine. Cet élément présente deux attributs :
     - Obligatoire : group identifie le groupe de capture. Une valeur comprise entre 0 (correspondance de séquence de texte totale) et 9 (groupes de correspondance toujours plus petits et ultérieurs) peut lui être attribuée
     - Facultatif : location indique une position dans le groupe de correspondance (par rapport au positionnement des parenthèses), start (parenthèse gauche) ou end (parenthèse droite).
   - Facultatif : <setFeature> crée une fonction et l'attribue à l'annotation. Cet élément présente deux attributs :
     - name est le nom de la fonction tel que vous l'avez défini dans la description du système de types

– `type` indique le type de la valeur de fonction ; il peut s'agir de String, Integer, Float et Reference. Le type doit être le même que le type de plage défini pour la fonction dans la description du système de types d'annotateurs.

  Les fonctions de type `Reference` sont utilisées pour créer un lien entre deux annotations pour modéliser une relation sémantique. Le contenu de l'élément `<setFeature>` doit être défini sur l'`id` de l'élément `<createAnnotation>` vers lequel vous voulez créer un lien.

Concepts associés

«Le fichier du jeu de règles», à la page 88

Personnalisation de l'annotateur d'expressions régulières

Vous pouvez personnaliser l'exemple de configuration de l'annotateur d'expressions régulières pour détecter de nouvelles entités (par exemple, des numéros de série de produits) ou adapter des règles d'expressions régulières pour les entrées existantes (par exemple, pour détecter des numéros de téléphone propres à une société en modifiant légèrement le jeu de règles et les fichiers du système de types.

Le fichier du jeu de règles et la description du système de types modifiés doivent être ajoutés au fichier archive du moteur de traitement d'expressions régulières (fichier PEAR). Une fois le fichier PEAR mis à jour, vous pouvez rajouter le moteur d'analyse de texte des expressions régulières personnalisé au système de recherche d'entreprise.

Pour une personnalisation plus poussée de l'annotateur d'expressions régulières, nous vous recommandons vivement d'utiliser les outils du SDK UIMA. Ces outils vous aident à créer ou à mettre à jour les fichiers de description du système de types et fichiers descripteurs, à éventuellement combiner l'annotateur à d'autres afin de former un moteur d'analyse global et de créer une nouvelle archive de moteur de traitement (fichier PEAR) incluant toutes les ressources nécessaires pour utiliser l'annotateur dans la recherche d'entreprise. Pour plus d'informations sur les outils disponibles pour vous aider à effectuer ces tâches, reportez-vous à la documentation du SDK UIMA.

Procédure

Pour adapter l'annotateur d'expressions régulières en ajoutant de nouvelles règles et entités ou pour modifier des règles existantes, vous pouvez mettre à jour l'exemple fourni de fichier PEAR de l'annotateur d'expressions régulières comme suit :

1. Créez un nouveau répertoire appelé `xml` dans votre système.
2. Copiez l'exemple de fichier de règles `of_sample_regex_rules.xml` du répertoire *ES_INSTALL_ROOT*`/packages/uima/regex/` vers votre répertoire `xml` et modifiez le fichier pour inclure vos règles de formes de correspondance personnalisées. Pour éviter des erreurs de syntaxe XML, utilisez un éditeur XML ou un outil de création XML de votre choix.
3. Copiez le fichier de description du système de types correspondant `of_sample_typesystem.xml` du répertoire `ES_INSTALL_ROOT/packages/uima/regex/` vers votre répertoire `xml` et modifiez le fichier pour inclure les définitions pour les types que vos nouvelles règles requièrent.
4. Si vous ajoutez seulement quelques nouvelles règles ou modifiez les règles existantes, vous n'êtes pas obligé de modifier le descripteur d'annotateur. Si

vous prévoyez d'apporter d'autres modifications ou si vous utilisez des étapes d'analyse personnalisée supplémentaires, vérifiez si le descripteur d'annotateur doit être modifié.

5. Utilisez un utilitaire d'archivage de votre choix pour mettre à jour une copie du fichier PEAR d'annotateur d'expressions régulières pour inclure vos deux fichiers mis à jour. Par exemple, copiez le fichier `of_regex.pear` de `ES_INSTALL_ROOT/packages/uima/regex/` vers le répertoire parent du répertoire `xml` que vous avez créé. Ensuite, utilisez l'outil de ligne de commande jar Java (par exemple, inclus dans le SDK IBM Java) pour générer les commandes suivantes à partir du répertoire parent :

   ```
   "jar -uf of_regex.pear -C xml/ of_sample_regex_rules.xml"
   "jar -uf of_regex.pear -C xml/ of_sample_regex_typesystem.xml"
   ```

6. Utilisez la console d'administration de la recherche d'entreprise pour ajouter l'annotateur d'expressions régulières en tant que moteur d'analyse de texte personnalisé au système de recherche d'entreprise et l'associer à une collection de documents test.
7. Vérifiez les résultats d'analyse générés par l'annotateur d'expressions régulières en mettant à jour les propriétés de la collection de documents pour produire une sortie XML lisible des résultats d'analyse stockés dans la structure d'analyse commune à l'aide de la fonction de cliché XCAS.
8. Traitez les documents test et utilisez le visualiseur d'annotations XCAS pour afficher le contenu des fichiers XML.
9. Si vous êtes satisfait des annotations créées par l'annotateur en fonction de vos expressions régulières personnalisées, modifiez à nouveau les propriétés de la collection de documents pour désactiver l'analyseur afin qu'il ne génère plus de sortie XML lisible des résultats d'analyse. Si d'autres modifications au fichier du jeu de règles sont requises, vous devez répéter les étapes de mise à jour du fichier PEAR.
10. Créez le fichier de mappage de la structure d'analyse commune à l'index pour indexer les résultats d'analyse ou le fichier de mappage de la structure d'analyse commune à la base de données si vous voulez ajouter les résultats à une base de données. Vous pouvez utiliser l'exemple fourni de fichier de mappage de la structure d'analyse commune à l'index comme point de départ pour créer votre fichier de mappage de la structure d'analyse commune à l'index.
11. Utilisation de la console d'administration de la recherche d'entreprise pour ajouter les fichiers de mappage et les associer à votre collection complète de documents.
12. Recherche de vos annotations à l'aide du fragment XML ou des requêtes XPath ou encore à l'aide de l'extension sémantique durant la recherche de synonymes.

Concepts associés

«Recherche sémantique facilitée à l'aide de l'annotateur d'expressions régulières», à la page 86

Tâches associées

«Affichage des résultats de l'annotateur de base et de l'analyse de texte personnalisée», à la page 13

Le descripteur d'annotateur

Le descripteur XML de l'annotateur d'expressions régulières contient les informations descriptives relatives à l'annotateur d'expressions régulières, requises pour exécuter l'annotateur.

Si vous utilisez simplement un annotateur d'expressions régulières et pas d'étape d'analyse personnalisée supplémentaire, vous ne devez modifier le descripteur que si :

- Vous voulez modifier le nom de fichier du fichier du jeu de règles (de l'élément `<externalResourceDependencies>`).
- Vous voulez utiliser plusieurs fichiers de jeux de règles.
- Vous voulez modifier le nom du fichier de description du système de types.

Si vous utilisez d'autres étapes d'analyse personnalisée, vous devez modifier le descripteur si :

- Vous voulez que votre analyse personnalisée utilise les annotations créées par l'annotateur d'expressions régulières. Dans ce cas, vous devez mettre à jour les fonctions de sortie du descripteur d'annotateur.
- Vous avez défini des règles d'expressions régulières qui doivent correspondre aux types d'annotation créés dans des étapes d'analyse personnalisée précédentes. Dans ce cas, vous devez mettre à jour les fonctions d'entrée du descripteur d'annotateur.

Utilisez les outils du SDK UIMA pour créer ou mettre à jour le descripteur d'annotateur et recréer l'archive du moteur de traitement (fichier .pear) incluant toutes les ressources requises pour utiliser l'annotateur dans la recherche d'entreprise. Pour plus d'informations sur les outils disponibles pour vous aider à effectuer ces tâches, reportez-vous à la documentation du SDK UIMA, à l'adresse http://www.alphaworks.ibm.com/tech/uima/.

Paramètres de configuration

L'annotateur d'expressions régulières présente un seul paramètre de configuration appelé`String2NumberImpl`, qui doit être défini sur le nom de la classe qui implémente l'interface `com.ibm.uima.an_regex.String2Number`. L'annotateur d'expressions régulières doit disposer d'une implémentation de cette classe, sinon une exception aura lieu. Si vous voulez personnaliser l'annotateur d'expressions régulières selon vos besoins, vous pouvez fournir votre propre implémentation de l'interface `String2Number` en passant votre nom de classe dans le fichier descripteur XML.

L'interface `String2Number` déclare deux méthodes, `toInt(String)` et `toFloat(String)`, qui transforment une représentation sous forme de chaîne d'un nombre entier ou d'une valeur flottante en un nombre entier ou une valeur flottante correspondante. Ces deux méthodes sont utilisées pour transformer un nombre contenant des caractères séparateurs en un nombre entier ou une valeur flottante Java valide.

L'implémentation par défaut de `com.ibm.uima.an_regex.String2Number_impl` considère un point (.) comme un séparateur décimal et une virgule (,) comme un séparateur de milliers. Par exemple, si 1,999.00 est trouvé dans un document texte, `toInt` le convertit en 1999. `toFloat` retourne 1999.00.

Exemple

La section du paramètre de configuration du descripteur est comme suit :

```
<configurationParameters>
 <configurationParameter>
  <name>String2NumberImpl</name>
  <description>Implémentation de l'interface
```

```
   com.ibm.uima.an_regex.String2Number</description>
  <type>String</type>
  <multiValued>false</multiValued>
  <mandatory>true</mandatory>
 </configurationParameter>

 <configurationParameterSettings>
  <nameValuePair>
   <name>String2NumberImpl</name>
   <value>
    <string>com.ibm.uima.an_regex.impl.String2Number_impl</string>
   </value>
  </nameValuePair>
 </configurationParameterSettings>
</configurationParameters>
```

Fonctions

Les fonctions d'entrée et de sortie de l'annotateur d'expressions régulières et les langues qu'il prend en charge sont définies dans la section des fonctions du descripteur de l'annotateur.

Les fonctions d'entrée (types d'entrée) du fichier descripteur doivent satisfaire les types de correspondance utilisés dans le fichier du jeu de règles. Si les règles utilisent uniquement le type `uima.tt.DocumentAnnotation`, vous n'avez pas à déclarer de fonctions d'entrée car ce type est toujours défini. Tous les autres types doivent être définis.

Les types d'annotation créés par l'annotateur d'expressions régulières sont spécifiés dans la section des fonctions de sortie. Ces types doivent correspondre aux types de sortie déclarés dans le fichier du jeu de règles.

Parce que l'annotateur d'expressions régulières est indépendant de la langue, indiquez `x-unspecified`, qui représente n'importe quelle langue.

Description du système de types

La section de la description du système de types du descripteur XML de l'annotateur d'expressions régulières définit le système de types utilisé par l'annotateur. Les types utilisés dans le fichier XML du jeu de règles et mentionnés dans les sections des fonctions d'entrée et de sortie du descripteur de l'annotateur doivent correspondre aux types définis dans la description du système de types.

Exemple

La section de la description du système de types du descripteur importe le fichier XML du descripteur du système de types :

```
<typeSystemDescription>
 <imports>
  <import location="./xml/of_sample_regex_typesystem.xml"/>
 </imports>
</typeSystemDescription>
```

Ressources externes

La section des ressources externes du descripteur contient les fichiers et classes requises par l'annotateur.

L'annotateur d'expressions régulières requiert le fichier du jeu de règles. Le fichier du jeu de règles est rendu disponible à l'annotateur d'expressions régulières à travers l'interface `com.ibm.uima.an_regex.FileResource`, implémentée par la classe `com.ibm.uima.an_regex.impl.FileResource_impl`. Pour passer vos règles personnalisées à l'annotateur d'expressions régulières, vous devez fournir le nom du fichier du jeu de règles dans le descripteur de l'annotateur et ajouter l'emplacement du fichier à votre chemin de classes. La clé que l'annotateur d'expressions régulières utilise pour accéder au fichier du jeu de règles s'appelle `RuleSetDefinition`. Ne modifiez pas cette clé, sinon l'annotateur d'expressions régulières ne trouvera pas le jeu de règles et l'annotateur ne pourra pas s'initialiser.

Les annotateurs personnalisés que vous déployez pour la recherche d'entreprise ne peuvent pas utiliser le paramètre UIMA `datapath` pour rechercher des ressources externes. Pour ce faire, spécifiez les chemins d'accès aux ressources dans le chemin d'accès aux classes de l'annotateur personnalisé. Pour savoir comment utiliser l'assistant de génération PEAR afin de définir les paramètres d'accès aux classes de l'annotateur personnalisé, consultez la documentation du SDK UIMA, à l'adresse http://www.alphaworks.ibm.com/tech/uima/.

Exemple

La section des ressources externes du descripteur est comme suit :

```
<externalResourceDependencies>
 <externalResourceDependency>
  <key>RuleSetDefinition</key>
  <description>Définition du jeu de règles</description>
  <interfaceName>com.ibm.uima.an_regex.FileResource</interfaceName>
  <optional>false</optional>
 </externalResourceDependency>
</externalResourceDependencies>
<resourceManagerConfiguration>
 <externalResources>
  <externalResource>
   <name>of_samples_regex_rules</name>
   <description>Fichier de définition du jeu de règles
                        pour les numéros de chambre</description>
   <fileResourceSpecifier>
    <fileUrl>file:of_samples_regex_rules.xml</fileUrl>
   </fileResourceSpecifier>
   <implementationName>
    com.ibm.uima.an_regex.impl.FileResource_impl</implementationName>
  </externalResource>
 </externalResources>
 <externalResourceBindings>
  <externalResourceBinding>
   <key>RuleSetDefinition</key>
   <resourceName>of_samples_regex_rules</resourceName>
  </externalResourceBinding>
 </externalResourceBindings>
</resourceManagerConfiguration>
```

Concepts associés

«Annotateur d'expressions régulières», à la page 85

«Le fichier du jeu de règles», à la page 88

Référence associée

«Journalisation», à la page 97

Journalisation

Tous les messages de journalisation provenant de l'annotateur d'expressions régulières sont écrits vers le fichier journal de la collection en cours.

Les fichiers journaux de collection se trouvent dans `ES_NODE_ROOT/logs/` et ont des noms au format `<collection_id>_<current_date>.log`. Vous pouvez visualiser les fichiers journaux au moyen des scripts esviewlogs.sh/.bat.

Il existe sept niveaux de journalisation possibles :

- Erreur
- Avertissement
- Info
- Config
- Fine
- Finer
- Finest

Le mappage des messages d'erreur et d'avertissement ne peut pas être modifié. Par défaut, seuls les messages d'information, d'avertissement et d'erreur sont écrits vers le fichier journal. Il s'agit des niveaux de journalisation standard utilisés par la recherche d'entreprise. Les autres niveaux de journalisation peuvent être mappés pour des informations plus détaillées.

Pour recevoir les messages de journalisation provenant de l'annotateur d'expressions régulières, le niveau de journalisation doit être défini au moins sur Config. A ce niveau, l'annotateur journalise les paramètres de configuration, tels que le fichier du jeu de règles utilisé et le nom de la classe d'implémentation de l'interface `com.ibm.uima.an_regex.String2Number`.

Si vous définissez le niveau de journalisation sur Finer par exemple, l'annotateur journalise les annotations qui n'ont pas pu être créées. Cela peut vous aider à déterminer la raison pour laquelle toutes les annotations prévues n'ont pas été créées. Par exemple, une erreur peut s'être glissée dans l'une de vos expressions régulières ou un groupe de capture facultatif peut ne pas avoir trouvé de correspondance dans le texte du document. De même, si vous indiquez qu'une fonction doit être définie sur la séquence de texte correspondant à un groupe de capture et qu'il n'y a pas de séquence de texte correspondante, la fonction est définie sur null.

Pour des informations très détaillées, définissez le niveau de journalisation sur Finest. A ce niveau, l'annotateur journalise le masque d'expression régulière en cours, la partie du texte du document qui est en cours d'analyse et toutes les annotations et fonctions qui ont été créées. L'utilisation de la journalisation très détaillée, notamment des niveaux Finer et Finest, a un impact négatif sur les performances globales de l'annotateur.

Si vous avez besoin d'un mappage avec un niveau de jouralisation détaillé, modifiez le fichier de configuration `tokenizer.properties` dans *ES_NODE_ROOT*`/master_config/parserservice/` en remplaçant le paramètre de configuration `trevi.tokenizer.jedii.InformationalLevelMapping=Info` par `trevi.tokenizer.jedii.InformationalLevelMapping=Finest`, par exemple.

Pour activer les modifications apportées au niveau de journalisation, vous devez arrêter tous les traitements d'analyseur syntaxique à l'aide de la console d'administration. Ensuite, vous devez arrêter, puis redémarrer la session du service d'analyseur syntaxique à partir de la ligne de commande en appelant :

```
>esadmin session parserservice stop
>esdamin session parserservice start
```

Après cela, l'analyse syntaxique pourra être redémarrée et vous devriez disposer du nouveau niveau de journalisation. Vous devez répéter ces étapes à chaque fois que vous modifiez le niveau de journalisation.

Concepts associés

«Annotateur d'expressions régulières», à la page 85

«Le fichier du jeu de règles», à la page 88

Référence associée

«Le descripteur d'annotateur», à la page 93

Documentation de la recherche d'entreprise

Vous pouvez consulter la documentation relative à OmniFind Enterprise Edition au format PDF ou HTML.

Le programme d'installation d'OmniFind Enterprise Edition installe automatiquement le centre de documentation, qui inclut une version de la documentation de recherche d'entreprise, au format html. En cas d'installation sur plusieurs serveurs, le centre de documentation est installé sur les deux serveurs de recherche. Si vous n'installez pas le centre de documentation, lorsque vous cliquez sur l'aide, le centre de documentation s'ouvre sur un site Web IBM.

Pour afficher les versions installées de la documentation PDF, accédez au répertoire ES_INSTALL_ROOT/docs/*environnement local*/pdf. Par exemple, pour rechercher les documents en anglais, accédez au répertoireES_INSTALL_ROOT/docs/en_US/pdf.

Pour accéder aux versions PDF de la documentation dans toutes les langues disponibles, consultez le site de la documentation OmniFind Enterprise Edition version 8.5.

Vous pouvez également accéder au téléchargement de produits, aux groupes de correctifs, aux notes techniques ainsi qu'au centre de documentation depuis le site OmniFind Enterprise Edition Support.

Le tableau ci-après indique la documentation disponible, les noms de fichier correspondants et l'emplacement de ces fichiers.

Tableau 12. Documentation pour la recherche d'entreprise

Titre	Nom de fichier	Emplacement
Centre de documentation		http://publib.boulder.ibm.com/infocenter/discover/v8r5/
Installation Guide for Enterprise Search	iiysi.pdf	ES_INSTALL_ROOT/docs/*environnement local*/pdf/
Quick Start Guide (Ce document est également disponible au format papier en anglais, français et japonais.)	OmniFindEE850_qsg_*deux lettres de l'environnement local*.pdf	ES_INSTALL_ROOT/docs/*environnement local*/pdf/
Administering Enterprise Search	iiysa.pdf	ES_INSTALL_ROOT/docs/*environnement local*/pdf/
Programming Guide and API Reference for Enterprise Search	iiysp.pdf	ES_INSTALL_ROOT/docs/en_US/pdf/
Troubleshooting Guide and Messages Reference	iiysm.pdf	ES_INSTALL_ROOT/docs/*environnement local*/pdf/
Text Analysis Integration	iiyst.pdf	ES_INSTALL_ROOT/docs/*environnement local*/pdf/
Plug-in pour Google Desktop Search	iiysg.pdf	ES_INSTALL_ROOT/docs/*environnement local*/pdf/

Fonctions d'accessibilité

Les fonctions d'accessibilité permettent aux utilisateurs qui présentent un handicap physique, comme une mobilité ou une vision réduites, d'utiliser pleinement les composants.

IBM s'efforce de proposer des produits accessibles à tous, quels que soient leur âge ou leurs capacités.

Fonctions d'accessibilité

La liste suivante comprend les fonctions d'accessibilité principales du produit OmniFind Enterprise Edition :

- Fonctionnement en mode clavier uniquement
- Interfaces utilisées par la plupart des lecteurs d'écran

Le centre de documentation OmniFind Enterprise Edition et les publications associées présentent des fonctions d'accessibilité avancées. Celles du centre de documentation sont décrites à la page suivante : http://publib.boulder.ibm.com/infocenter/discover/v8r5m0/topic/com.ibm.classify.nav.doc/dochome/accessibility_info.htm.

Navigation via le clavier

Ce produit utilise les touches de navigation Microsoft Windows standard.

Les touches de raccourci suivantes permettent également de se déplacer et d'avancer dans le programme d'installation d'OmniFind Enterprise Edition.

Tableau 13. Touches de raccourci du programme d'installation

Action	Raccourci
Mettre en évidence un bouton d'option	Touche de déplacement
Sélectionner un bouton d'option	Touche de tabulation
Mettre en évidence un bouton de fonction	Touche de tabulation
Sélectionner un bouton de fonction	Touche Entrée
Accéder à la fenêtre suivante ou précédente ou annuler	Mettre en évidence un bouton de fonction en appuyant sur la touche de tabulation et sur la touche Entrée
Rendre la fenêtre active inactive	Ctrl + Alt + Echap

Informations de l'interface

Les interfaces utilisateur correspondant à la console d'administration, à l'exemple d'application de recherche et au personnaliseur d'exemple d'application de recherche sont basées sur des navigateurs ; vous pouvez les afficher via Microsoft Internet Explorer ou Mozilla FireFox. Pour obtenir une liste des touches de raccourci et des autres fonctions d'accessibilité de votre navigateur, reportez-vous à l'aide en ligne d'Internet Explorer ou de FireFox.

Informations complémentaires sur l'accessibilité

Vous pouvez afficher les publications relatives à OmniFind Enterprise Edition au format PDF (Adobe Portable Document Format), via le logiciel Adobe Acrobat Reader. Ces documents PDF apparaissent sur le CD accompagnant le produit, mais vous pouvez également les télécharger sur le site suivant : http://www.ibm.com/support/docview.wss?rs=63&uid=swg27010938.

IBM et l'accessibilité

Pour en savoir plus sur l'engagement d'IBM en matière d'accessibilité, consultez la page IBM Human Ability and Accessibility Center (en anglais).

Glossaire des termes de la recherche d'entreprise

Ce glossaire définit les termes utilisés dans les interfaces et la documentation de la recherche d'entreprise.

administrateur de recherche d'entreprise
Rôle administratif permettant à un utilisateur d'administrer l'intégralité du système de recherche d'entreprise.

adresse IP
Adresse unique pour un périphérique ou une unité logique utilisant la norme IP.

adresse URL
Adresse unique d'une ressource d'information accessible dans un réseau comme Internet. L'URL inclut l'abréviation du protocole utilisé pour accéder à la ressource d'information et les informations utilisées par le protocole pour la rechercher.

adresse URL de départ
Point de départ d'une exploration

affinité lexicale
Relation entre les mots à rechercher dans un document qui ont un sens similaire. L'affinité lexicale permet de déterminer la pertinence d'un résultat.

agent d'utilisateur
Application parcourant le Web et laissant des informations sur elle-même sur les sites qu'elle visite. Dans la recherche d'entreprise, le moteur de balayage du Web est un agent utilisateur.

analyse des liens
Méthode reposant sur l'analyse des hyperliens entre les documents et permettant de déterminer les pages de la collection importantes pour les utilisateurs.

analyse de texte
Processus d'extraction de la sémantique et d'autres informations du texte pour améliorer les possibilités d'extraction des données d'une collection. Voir aussi recherche sémantique.

analyseur syntaxique
Programme interprétant les documents ajoutés au magasin de données de la recherche d'entreprise. L'analyseur syntaxique extrait les informations des documents et les prépare pour indexation, recherche et extraction.

annotateur
Composant de logiciel qui effectue des tâches d'analyse linguistique spécifiques et qui génère et enregistre des annotations. Un annotateur correspond au composant de la logique d'analyse d'un moteur d'analyse.

annotateur d'expression régulière
Composant logiciel détectant des entités ou des unités d'informations dans des documents texte, tels que des numéros de produits, à partir d'expressions régulières qui décrivent les modèles exacts recherchés dans les documents texte. Si l'une des expressions régulières correspond à des parties du texte du document, l'annotateur d'expression régulière crée les

annotations correspondantes qui couvrent en partie ou en totalité l'élément concerné. Ces expressions annotées sont ensuite stockées dans l'index de recherche d'entreprise à l'aide d'un fichier de mappage d'index ou dans une base de données compatible JDBC à l'aide d'un fichier de mappage de base de données.

annotateurs de base de la recherche d'entreprise
Ensemble de moteurs d'analyse de texte standard utilisé dans la recherche d'entreprise pour le traitement de l'analyse de documents par défaut.

annotation
Informations sur une étendue de texte. Par exemple, une annotation peut indiquer qu'une étendue de texte représente un nom de société. Dans l'architecture UIMA, une annotation correspond à un type spécifique de structure de fonctions.

application de recherche
Dans le domaine de la recherche d'entreprise, programme qui traite les requêtes, effectue des recherches dans l'index, renvoie les résultats et extrait les documents source.

arborescence des catégories
Hiérarchie des catégories.

archive du moteur de traitement
Fichier d'archive zip .pear incluant un moteur d'analyse UIMA et toutes les ressources requises pour l'utiliser pour une analyse personnalisée dans la recherche d'entreprise.

autorité de certification
Organisation ou société tierce digne de confiance qui émet les certificats numériques utilisés pour créer des signatures numériques et des paires de clés publique-privée. L'autorité de certification garantit l'identité des personnes à qui le certificat unique est accordé.

bibliothèque
Objet système servant de répertoire à d'autres objets. Voir aussi bibliothèque Domino Document Manager.

bibliothèque Domino Document Manager
Base de données Domino Document Manager servant de point d'entrée à Domino Document Manager.

caractère d'échappement
Caractère supprimant ou sélectionnant une signification spéciale d'un ou plusieurs caractères qui suivent.

caractère de fin
Caractère occupant la dernière position d'un mot.

caractère de masquage
Caractère permettant de représenter les caractères facultatifs au début, au milieu et à la fin d'un terme à rechercher. Les caractères de masquage sont généralement utilisés pour rechercher des variations d'un terme dans un index. Voir aussi caractère générique.

caractère d'interligne
Caractère de contrôle entraînant le déplacement d'une ligne vers le bas d'une position d'impression ou d'affichage.

caractère générique
Caractère permettant de représenter les caractères facultatifs au début, au milieu ou à la fin d'un terme à rechercher.

catégorie à base de règles
Catégorie créée par des règles qui indiquent quels documents sont associés à quelles catégories. Par exemple, vous pouvez définir des règles pour associer à des catégories spécifiques des documents qui contiennent ou excluent certains mots ou qui correspondent à un masque d'URI.

certificat
En termes de sécurité informatique, document numérique associant une clé publique à l'identité du propriétaire du certificat et permettant ainsi l'authentification du propriétaire. Un certificat est émis par une autorité de certification qui le signe numériquement.

chemin d'accès à la fonction
Chemin permettant d'accéder à la valeur d'une fonction dans une structure de fonctions UIMA.

classe de pondération
Objet contenant des spécifications pouvant influencer le classement relatif d'un document dans les résultats de la recherche.

classement
Affectation d'un entier à chaque document des résultats de la recherche d'une requête. L'ordre des documents dans les résultats de la recherche repose sur la pertinence de la requête. Un rang plus élevé correspond à une occurrence plus proche. Voir aussi classement dynamique et classement statique.

classement dynamique
Type de classement dans lequel les termes de la requête sont analysés par rapport aux documents explorés pour déterminer le rang des résultats. Voir aussi évaluation textuelle des données. Par opposition à classement statique.

classement populaire
Type de classement qui modifie le classement existant d'un document en fonction de la popularité de ce dernier.

classement statique
Type de classement dans lequel certains facteurs des documents à classer (date, nombre de liens à ces documents et autres) augmentent le rang des documents. Par opposition à classement dynamique.

clitique
Mot fonctionnant de manière distincte du point de vue de la syntaxe, mais qui est connecté phonétiquement à un autre mot. Un clitique peut être écrit comme attaché au mot auquel il est lié ou détaché de ce mot. Par exemple, en français, les articles définis ou indéfinis (tels que ***le*** ou ***un***) sont des clitiques.

collection
Ensemble de sources de données et d'options permettant d'explorer, d'analyser, d'indexer ces sources de données et d'y effectuer des recherches.

connectivité JDBC
Norme informatique pour une connectivité indépendante de la base de

données entre la plateforme Java et une plage étendue de bases de données. L'interface JDBC fournit une API au niveau des appels pour l'accès aux bases de données SQL.

consommateur de la structure d'analyse commune (CAS)
Consommateur effectuant le traitement final sur les résultats d'analyse stockés dans la structure d'analyse commune. Par exemple, un consommateur indexe le contenu de la structure d'analyse commune dans un moteur de recherche ou charge des résultats d'analyse spécifiques dans une base de données relationnelle.

couche CCL (common communication layer)
Infrastructure de communication unissant les divers composants (contrôleur, analyseur syntaxique, moteur de balayage, serveur d'index) d'OmniFind Enterprise Edition.

coupure
Voir coupure de mot.

coupure de mot
Processus de normalisation linguistique au cours duquel les variantes d'un mot sont réduites à une forme commune. Par exemple, les mots tels que *connexions*, *connecteur* et *connecté* sont réduits au terme *conn*.

dictionnaire des synonymes
Dictionnaire permettant aux utilisateurs de rechercher des synonymes des termes de leur requête lorsqu'il effectue une recherche dans une collection.

DIIOP (Domino Internet Inter-ORB Protocol)
Tâche serveur exécutée sur le serveur et fonctionnant avec la fonction ORB (Object Request Broker) de Domino pour permettre les communications entre les applets Java créées avec les classes Java Notes et le serveur Domino. Les utilisateurs de navigateur et les serveurs Domino utilisent DIIOP pour communiquer et échanger des données d'objet.

Document Object Model
Système dans lequel un document structuré, tel qu'un fichier XML, est affiché comme arborescence d'objets accessibles et pouvant être mis à jour à l'aide d'un programme.

dossier Domino Document Manager
Base de données Domino Document Manager permettant d'organiser des documents. Les dossiers contiennent des bases de données Domino.

droit d'accès
Informations détaillées, acquises lors de l'authentification, qui décrivent l'utilisateur, les associations de groupes et autres attributs d'identité relatifs à la sécurité. Les droits d'accès permettent d'effectuer une multitude de services, tels que les fonctions d'autorisation, d'audit et de délégation. Par exemple, les informations d'identification (ID et mot de passe utilisateur) d'un utilisateur lui permettent d'accéder à un compte.

emplacement
Programme permettant aux utilisateurs de créer des documents lisibles par d'autres utilisateurs. Ces derniers pourront y répondre, les commenter, mais également vérifier le statut et les échéances du projet. Les utilisateurs peuvent également discuter avec d'autres utilisateurs qui se trouvent au même emplacement. Voir aussi emplacement Lotus QuickPlace.

emplacement Lotus QuickPlace
Zone partitionnée d'un espace Lotus QuickPlace restreinte aux membres autorisés qui partagent un intérêt commun et doivent travailler ensemble.

espace Emplacement virtuel visible dans le portail, où des utilisateurs et des groupes se rencontrent pour collaborer. Dans un portail, chaque utilisateur dispose d'un espace personnel pour son travail privé et les utilisateurs et les groupes ont accès à divers espaces partagés, publics ou à accès restreint. Voir aussi espace Lotus QuickPlace.

espace d'exploration
Ensemble de sources correspondant aux masques spécifiés (tels que des adresses URL, des noms de base de données, des chemins de système de fichiers, des noms de domaine et des adresses IP) qu'un moteur de balayage explore pour extraire des éléments à indexer.

espace Lotus QuickPlace
Emplacement Web fourni par Lotus QuickPlace qui permet à des participants dispersés géographiquement de collaborer sur des projets et de communiquer en ligne dans un espace de travail structuré et sécurisé.

évaluation basée sur le texte
Processus d'affectation d'un entier à un document pour indiquer la pertinence du document par rapport aux termes d'une requête. Plus cet entier est élevé, plus l'occurrence est proche de la requête. Voir aussi classement dynamique.

extraction de concept
Fonction d'analyse de texte identifiant les éléments de vocabulaire importants (tels que les personnes, les endroits ou les produits) dans les documents texte et générant une liste de ces éléments. Voir aussi extraction de thème.

extraction de thème
Type d'extraction de concept qui reconnaît automatiquement les éléments de vocabulaire importants dans les documents texte pour extraire le thème ou le sujet d'un document. Voir aussi extraction de concept.

extraction d'informations
Type d'extraction de concept qui reconnaît automatiquement les éléments de vocabulaire importants, tels que les noms, les termes et les expressions dans les documents de texte.

fédérateur distant
Fédérateur serveur qui regroupe un ensemble d'objets pouvant être explorés.

fédérateur local
Dans une application de recherche d'entreprise, objet client créé par les API de recherche et d'indexation permettant aux utilisateurs d'effectuer des recherches dans un ensemble de collections hétérogènes et d'obtenir un ensemble unifié de résultats.

fédération
Processus de combinaison de systèmes d'affectation de nom pour que le système obtenu puisse traiter les noms composites qui couvrent les systèmes d'affectation de nom.

fichier de base de données de clés
Voir jeu de clés.

fichier du magasin de clés
Jeu de clés contenant des clés publiques stockées sous forme de certificats de signataire ainsi que des clés privées stockées dans des certificats personnels.

fichiers d'index de recherche
Ensemble de fichiers dans lequel un index est stocké dans le moteur de recherche.

file d'attente d'index
Liste de demandes de génération d'index principaux et delta.

génération de l'index principal
Dans un système de recherche d'entreprise, processus de génération d'un index entier. Par opposition à génération d'index delta.

génération d'index delta
Dans un système de recherche d'entreprise, processus d'ajout de nouvelles informations à un index existant. Par opposition à génération de l'index principal.

gestion des identités
Ensemble d'API de recherche d'entreprise contrôlant l'accès à des données sécurisées et permettant aux utilisateurs d'effectuer une recherche dans une collection sans avoir à indiquer un ID et mot de passe utilisateur pour chaque référentiel de la collection.

hachage
Chaîne de jetons (mots) consécutifs tirés d'une phrase. Par exemple, pour ″Cette phrase est une phrase courte″, les hachages de trois mots (ou les trigrammes) sont :

Cette phrase est phrase est une
est une phrase
une phrase courte

Les hachages peuvent être utilisés en linguistique statistique. Par exemple, si deux textes ont de nombreux hachages communs, il existe certainement un lien entre ces textes.

identificateur URI
Chaîne de caractères compacte identifiant une ressource abstraite ou physique.

identification de la langue
Dans la recherche d'entreprise, fonction déterminant la langue d'un document.

index Voir index de recherche.

index de recherche
Structure de données faisant référence à des éléments de données pour permettre à une fonction de recherche de détecter les documents qui contiennent les termes de la requête.

instruction nofollow
Instruction dans une page Web qui indique aux robots (tels que le moteur de balayage Web) de ne pas suivre les liens détectés dans cette page.

instruction noindex
Instruction dans une page Web qui indique aux robots (tels que le moteur de balayage du Web) de ne pas inclure le contenu de cette page dans l'index.

JavaScript
Langage de script Web utilisé dans les navigateurs et les serveurs Web.

jeton de sécurité
Informations sur l'identité et la sécurité permettant d'autoriser l'accès aux documents d'une collection. Des types de source de données différents prennent en charge des types de jeton de sécurité différents. Les rôles utilisateur, ID utilisateur, ID groupe et autres informations permettant de contrôler l'accès au contenu en sont des exemples.

jeu de clés
En termes de sécurité informatique, fichier contenant des clés publiques et privées, des clés d'authentification et des certificats. Voir aussi fichier du magasin de clés.

katakana
Jeu de caractères composé de symboles utilisés dans l'un des deux alphabets phonétiques japonais courants, utilisé principalement pour écrire des mots étrangers phonétiquement.

langage XPath (XML Path)
Langage conçu pour identifier ou traiter de manière unique des éléments de données XML source, pour une utilisation avec des technologies XML, telles que les analyseurs syntaxiques XSLT, XQuery, et XML. XPath est une norme du consortium World Wide Web.

LDAP (Lightweight Directory Access Protocol)
Protocole ouvert utilisant TCP/IP pour permettre l'accès aux répertoires qui prennent en charge un modèle X.500 et qui ne requiert pas une quantité de ressources aussi importante que le protocole DAP (Directory Access Protocol) X.500 plus complexe. Par exemple, LDAP permet de localiser des personnes, des entreprises et d'autres ressources dans un répertoire Internet ou intranet.

lemmatisation
Processus identifiant la racine et différentes formes grammaticales d'un mot. Par exemple, une recherche du mot ″chat″ renvoie des documents contenant également le mot ″chats″. De même une recherche du mot ″partir″ renvoie également des documents contenant les mots ″partant″, ″parti″ ou ″partait″.

lemme
Forme de base d'un mot. Les lemmes sont importants dans les langues utilisant beaucoup les déclinaisons, telles que le tchèque.

ligature
Plusieurs caractères reliés de sorte qu'il apparaissent comme un même caractère. Par exemple, les caractères ff et ffi peuvent être présentés comme des ligatures.

liste de contrôle d'accès (ACL)
En termes de sécurité informatique, liste associée à un objet qui identifie tous les sujets qui peuvent y accéder et leurs droits d'accès.

logiciel de recherche
Fonction d'un moteur de balayage qui détermine les sources de données desquelles le moteur de balayage peut extraire des informations.

machine virtuelle Java (JVM)
Implémentation logicielle d'un processeur qui exécute du code Java (applets et applications) compilé.

magasin de données
Structure de données dans laquelle les documents sont conservés sous leur forme analysée.

magasin de données brutes
Structure de données dans laquelle les documents explorés sont stockés avant d'être envoyés à l'analyseur syntaxique. Les moteurs de balayage écrivent dans le magasin de données et brutes et l'analyseur syntaxique lit ces données. Une fois explorés, les documents sont retirés du magasin de données brutes. A ne pas confondre avec magasin de données.

marquage sémantique
Processus consistant à analyser les entrées pour en faire des jetons.

marqueur sémantique
Programme de segmentation du texte qui analyse ce dernier et détermine si et quand une série de caractères peut être reconnue comme symbole.

mémoire cache de recherche
Mémoire tampon conservant les données et les résultats des demandes de recherche précédentes.

mettre en file d'attente
Mettre un message ou un élément dans une file d'attente.

moniteur
Utilisateur de la recherche d'entreprise disposant des droits requis pour observer des processus au niveau des collections.

mot avec degré de pondération
Mot pouvant influencer le classement relatif d'un document dans les résultats de la recherche. Lors du traitement de la requête, l'importance d'un document qui contient un mot avec degré de pondération peut être augmentée ou réduite, en fonction d'un score prédéfini pour ce mot.

moteur d'analyse
Voir moteur d'analyse de texte.

moteur d'analyse de texte
Composant logiciel chargé de rechercher et de représenter le contexte et le contenu sémantique dans le texte.

moteur d'analyse de texte personnalisé
Moteur d'analyse de texte créé à l'aide du kit de développement de logiciels (SDK) de l'architecture UIMA et pouvant être ajouté à l'ensemble des moteurs d'analyse de texte de recherche d'entreprise (également connus sous le nom de d'annotateurs de base de la recherche d'entreprise). Voir aussi moteur d'analyse de texte.

moteur de balayage
Logiciel qui extrait les documents des sources de données et collecte des informations permettant de créer des index de recherche.

moteur de balayage du Web
Type de moteur de balayage qui explore le Web en extrayant un document Web et en suivant les liens de ce document.

moteur de recherche
Programme acceptant une demande de recherche et renvoyant une liste des documents à l'utilisateur.

mot vide
Mot couramment utilisé, tel que *le*, *un* ou *et*, ignoré par une application de recherche.

nom distinctif
Nom identifiant de manière unique une entrée dans un répertoire. Un nom distinctif est constitué de paires attribut:valeur, séparées par des virgules. Egalement un ensemble de paires nom-valeur (telles que CN=nom de la personne et C=pays ou région) qui identifie de manière unique une entité dans un certificat numérique.

normalisation des caractères
Processus au cours duquel les variantes d'un caractère, telles que les marques de mise en majuscules et les marques diacritiques sont réduites à une forme commune.

NRPC (Notes remote procedure call)
Mécanisme de communication de Lotus Notes utilisé pour toutes les communications Notes à Notes.

opérateur
Utilisateur de la recherche d'entreprise disposant des droits requis pour observer, démarrer et arrêter des processus au niveau des collections.

page de liste de valeurs de départ
Dans WebSphere Portal, page XML qui contient les liens d'accès aux pages disponibles sur un portail. Les moteurs de balayage utilisent cette liste pour identifier les documents à explorer. La page contient également des métadonnées stockées avec les documents explorés dans l'index de recherche d'entreprise.

page d'erreur temporaire
Type de page Web qui fournit des informations sur la raison pour laquelle la page Web ne peut être renvoyée. Par exemple, au lieu de renvoyer un simple code d'état, le serveur HTTP renvoie une page expliquant le code d'état en détail.

pages JSP (JavaServer Page)
Technologie de scriptage serveur permettant l'imbrication dynamique de code Java dans des pages Web (fichiers HTML) et l'exécution de ce code lorsque la page est utilisée, pour renvoyer un contenu dynamique à un client.

pilote d'analyseur syntaxique
Dans un système de recherche d'entreprise, service fournissant des documents au service de l'analyseur syntaxique. Il existe un pilote d'analyseur syntaxique pour chaque collection. Le pilote d'analyseur syntaxique de la collection correspond à son analyseur syntaxique dans la console d'administration de la recherche d'entreprise.

Portal Document Manager (PDM)
Permet aux utilisateurs d'avoir un référentiel de documents central pour

faciliter la collaboration entre équipes. Les administrateurs peuvent gérer efficacement leurs documents et contrôler l'interaction des utilisateurs avec les informations.

protocole d'exclusion de robots
Protocole permettant aux administrateurs de site Web d'indiquer aux robots qui visitent leur site à quelle partie de ce dernier ils ne doivent pas accéder.

raccourci de lien
Association entre un identificateur URI et des mots clés ou expressions.

récapitulatif
Processus d'insertion de phrases non redondantes dans les résultats de la recherche pour décrire brièvement le contenu d'un document. Voir aussi regroupement dynamique et regroupement statique.

recherche booléenne
Recherche dans laquelle un ou plusieurs termes sont combinés à l'aide d'opérateurs tels que AND, NOT et OR.

recherche de proximité
Recherche de texte renvoyant un résultat lorsque deux ou plusieurs termes correspondant sont placés à une certaine distance l'un de l'autre, comme par exemple dans la même phrase ou le même paragraphe.

recherche de termes pondérée
Requête dans laquelle certains termes sont plus importants que d'autres.

recherche de texte libre
Recherche dans laquelle le terme à rechercher est exprimé sous forme de texte libre.

recherche fédérée
Fonction de recherche permettant les recherches sur plusieurs services de recherche et renvoyant une liste consolidée de résultats de recherche.

recherche floue
Recherche renvoyant des mots dont l'orthographe est similaire à celle du terme à rechercher.

recherche hybride
Recherche combinée d'éléments booléens et de texte libre.

recherche linguistique
Type de recherche qui explore, extrait et indexe un document avec des termes réduits à leur forme de base (par exemple, pour que *animaux* soit indexé sous *animal*) ou développés avec leur forme de base (comme pour les mots composés).

recherche paramétrique
Type de recherche portant sur les objets qui contiennent une valeur ou un attribut numérique, tel que des dates, des entiers ou d'autres types de données numériques dans une plage spécifique.

recherche par zone
Requête restreinte à une zone particulière.

recherche sémantique
Type de mot clé intégrant l'analyse linguistique et contextuelle. Voir aussi analyse de texte.

regroupement dynamique
Type de regroupement dans lequel les termes de la recherche sont mis en évidence et les résultats de la recherche contiennent des phrases qui représentent au mieux les concepts du document que l'utilisateur recherche. Par opposition à regroupement statique.

regroupement statique
Type de regroupement dans lequel les résultats de la recherche contiennent un récapitulatif stocké et spécifié du document. Par opposition à regroupement dynamique.

résultats d'analyse
Informations générées par les annotateurs. Les résultats d'analyse sont enregistrés dans une structure de données appelée structure d'analyse commune. Les résultats d'analyse fournis par les moteurs d'analyse de texte personnalisés (annotateurs) peuvent être inclus dans l'index de recherche d'entreprise.

résultats de la recherche
Liste de documents correspondant à la demande de recherche.

retirer d'une file d'attente
Supprimer des éléments d'une file d'attente.

rôle administratif
Classification d'un utilisateur qui octroie l'accès à un utilisateur.

segmentation
Division du texte en unités lexicales distinctes. Le traitement sans dictionnaire comprend les espaces entre les mots et la segmentation n-gram, tandis que le traitement avec dictionnaire comprend la segmentation de mots, de phrases et de paragraphes ainsi que la lemmatisation.

segmentation de texte
Voir segmentation.

segmentation n-gram
Méthode d'analyse qui considère les séquences de chevauchement d'un nombre de caractères donné comme un même mot au lieu de délimiter les mots par des espaces comme dans la segmentation Unicode des espaces.

segmentation Unicode des espaces
Méthode de marquage sémantique utilisant les propriétés des caractères Unicode pour distinguer les marques des séparateurs.

serveur proxy
Serveur agissant comme intermédiaire pour les demandes Web HTTP hébergées par une application ou un serveur Web. Un serveur proxy se substitue aux serveurs de contenu de l'entreprise.

service d'analyseur syntaxique
Service de recherche d'entreprise gérant l'analyse syntaxique de tous les documents et le traitement de l'analyse de texte dans toutes les collections de documents. Au moins un service d'analyseur syntaxique fonctionne en permanence.

servlet
Programme Java exécuté sur un serveur Web, qui étend les fonctionnalités du serveur en générant un contenu dynamique en réponse aux demandes des clients Web. Les servlets sont généralement utilisés pour connecter des bases de données au Web.

signe diacritique
Signe indiquant un changement de valeur phonétique d'un caractère ou d'une combinaison de caractères.

source de données
Tout référentiel de données dont il est possible d'extraire des documents (par exemple, les bases de données Web, relationnelles, non relationnelles et les systèmes de gestion de contenu).

source de données externe
Source de données d'une fédération non explorée, analysée ou indexée par OmniFind Enterprise Edition. Les recherches des sources de données externes sont déléguées à l'interface de programmation d'application des requêtes de ces sources de données.

SSL (Secure Sockets Layer)
Protocole de sécurité permettant la confidentialité des communications. Avec SSL, les applications client/serveur peuvent communiquer sans risque d'écoute clandestine, de contrefaçon et de falsification des messages.

structure d'analyse commune (CAS)
Structure stockant le contenu et les métadonnées d'un document ainsi que tous les résultats d'analyse fournis par un moteur d'analyse de texte. Toutes les données d'échange au cours de l'analyse des documents sont gérées par l'utilisation de la structure d'analyse commune.

structure de fonctions
Structure de données sous-jacente représentant le résultat de l'analyse de texte. Une structure de fonctions est une structure attribut-valeur. Chaque structure de fonctions appartient à un certain type et chaque type possède un ensemble de fonctions ou d'attributs valides, similaire à une classe Java.

suppression des mots vides
Processus de suppression des mots vides de la requête pour ignorer les mots courants et renvoyer des résultats plus pertinents.

symbole
Unités textuelles de base indexées par la recherche d'entreprise. Les symboles peuvent correspondre aux mots d'une langue ou à d'autres unités de texte appropriées pour l'indexation.

système de types
Le système de types définit les types d'objets (structures de fonctions) pouvant être découverts par un moteur d'analyse de texte dans un document. Il définit toutes les structures de fonctions possibles en termes de types et de fonctions. Vous pouvez définir autant de types différents que vous le souhaitez dans un système de types. Un système de types est spécifique à un domaine et à une application.

taxonomie
Classification des objets en groupes en fonction de leurs similitudes. Dans la recherche d'entreprise, une taxonomie organise les données en catégories et sous-catégories. Voir aussi arborescence des catégories.

texte libre
Texte non structuré constitué de mots ou phrases.

type de MIME
Norme Internet permettant d'identifier le type d'objet transféré sur Internet.

type de source de données
Regroupement de sources de données en fonction du protocole utilisé pour accéder aux données.

UIMA (Unstructured Information Management Architecture)
Architecture IBM qui définit une structure d'implémentation des systèmes pour l'analyse des données non structurées.

zone Zone destinée à la saisie d'une catégorie spécifique de données ou d'informations de contrôle.

Remarques et marques

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays.

Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils

contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins

illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programme d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. _entrez la ou les années_. All rights reserved.

Voici des parties de ce produit :

- Oracle® Outside In Content Access, Copyright © 1992, 2008, Oracle. All rights reserved.
- IBM XSLT Processor Eléments sous licence - Propriété d'IBM ©Copyright IBM Corp., 1999-2008. All Rights Reserved.

Marques

Pour plus d'informations sur les marques IBM, voir http://www.ibm.com/legal/copytrade.shtml.

Les termes qui suivent sont des marques d'autres sociétés :

Adobe, Acrobat, Portable Document Format (PDF), PostScript et toutes les marques d'Adobe sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou certains autres pays.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques d'Intel Corporation ou de ses filiales aux Etats-Unis et dans certains autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Index

U

V

SC11-2398-02