فصلنامهٔ مطالعات مدیریت کسب و کار هوشمند- سال ششم ـ شماره ۲۴ـ تابستان ۱۳۹۷
صفحات ۶۲-۳۷

روشی جدید برای خوشه‌بندی اسناد HTML با استفاده از الگوریتم‌های تلفیقی

مریم شعار *
علی‌اصغر سالارنژاد **

چکیده

با عنایت به حجم بالای اطلاعات کنونی وب توجه به سیستم‌های خودکار استخراج اطلاعات بیشتر شده است. از مهم‌ترین روش‌های خودکار استخراج اطلاعات، خوشه‌بندی می‌باشد. روش‌های خوشه‌بندی زیادی تابه‌حال ارائه شده است که اکثراً مبتنی بر مدل برداری می‌باشند. در این مدل با هر سند مانند مجموعه‌ای از کلمات برخورد می‌گردد و توالی کلمات در جمله، نادیده گرفته می‌شود. ازآنجایی‌که معانی در زبان طبیعی به‌طور کامل وابسته به توالی کلمات می‌باشند نقیصه بزرگی در این روش‌ها احساس می‌گردد. برای رفع این نقیصه در این مقاله روشی جدید در خوشه‌بندی اسناد Html ارائه گردیده است که در آن الگوریتم Stc برای خوشه‌بندی Snippetها لحاظ شده‌است. این روش که با عنوان خوشه‌بندی بر اساس جملات کلیدی Ks_Stc مطرح شده برای هر سند بردار وزن‌داری تهیه می‌کند و با استفاده از این بردار، جملات کلیدی هر متن از سند استخراج می‌گردد و نهایتاً این جملات کلیدی برای خوشه‌بندی به الگوریتم Stc داده می‌شود.

کلیدواژگان: افزونگی اطلاعات، خوشه‌بندی اسناد Html، داده‌کاوی، سیستم‌های استخراج اطلاعات، کلاس‌بندی.

* استادیار، گروه مدیریت صنعتی، دانشکده مدیریت، دانشگاه آزاد اسلامی، واحد تهران شمال، تهران.(نویسنده مسئول)
maryam.shoar@gmail.com
** کارشناسی ارشد، مدیریت فناوری اطلاعات، دانشکده مدیریت، دانشگاه آزاد اسلامی، واحد تهران شمال، تهران.

تاریخ دریافت:۱۳۹۵/۱۰/۱۵ **تاریخ پذیرش:۱۳۹۶/۰۴/۲۷**

مقدمه

بر اساس مطالعات انجام‌گرفته در سال ۲۰۰۵ بیش از ۱۱,۵ بیلیون صفحه در وب جهانی ایندکس شده است (گولی و سینگورینی[1]، ۲۰۰۵). پایه اصلی این صفحات را اسناد Html تشکیل می‌دهند. این‌گونه اسناد از نوع نیمه ساخت‌یافته تلقی می‌گردند. متون ساخت‌یافته شامل پایگاه‌های داده متنی و یا اسناد متنی می‌گردد که استاندارد معینی در ساختار آن‌ها رعایت شده است و معمولاً استخراج اطلاعات از آن‌ها با کمک این ساختار معین به سهولت انجام می‌شود. اسناد متنی نیمه ساخت‌یافته در نقطه‌ای میان اسناد ساخت‌یافته و غیر ساخت‌یافته قرارگرفته‌اند. به‌عبارت‌دیگر این‌گونه اسناد ممکن است از اصول معین‌شده در ساختارشان تبعیت نکنند و با ساختارهای متفاوت طراحی شده باشند (آزاد و آبشیک[2]، ۲۰۱۴). از طرفی این اسناد علاوه بر متن شامل تصویر و صوت در فرمت‌های متفاوت نیز می‌باشند. حجم اطلاعات موجود در این فرمت تا حدی است که مثلاً درباره کلمه ساده‌ای مثل "Iran" یک حجم ترابایتی از اطلاعات را می‌توان یافت. در مواجهه با این‌گونه مشاهدات عبارت افزونگی اطلاعات مطرح می‌گردد. این واقعیت که اطلاعات موجود در وب جهانی به‌صورت دینامیک و با روند رو به رشدی در حال افزایش است نیز چالش دیگری در این زمینه است به‌نحوی‌که از وب جهانی با نام "بزرگ‌ترین پایگاه دانش " یاد شده است (فررا و همکاران[3]، ۲۰۱۴) این موارد ما را وادار به پرسیدن این سؤال می‌کند که "آیا امکان استخراج اطلاعات موردعلاقه افراد از این حجم بالای رو به افزایش وجود دارد؟". در حال حاضر دو راه برای دستیابی به اطلاعات موردنیاز در وب جهانی موجود است: اول از طریق مرور دستی و متوالی صفحات به‌هم پیوسته وب تا کاربر به اطلاعات موردنظر خود دست یابد و دوم از طریق موتورهای جستجو و با وارد کردن یک کلمه کلیدی مرتبط با موضوع. شاید روش دوم در ابتدا امیدبخش باشد اما در واقع کاربران این موتورها با لیست مرتب‌شده‌ای از یافته‌ها مواجه هستند که کشف اطلاعات موردنظرشان در آن نیز نیازمند صرف وقت و انرژی بالایی است. جهت رفع این مشکل جامعه استخراج اطلاعات به مسئله داده‌کاوی و علی‌الخصوص یکی از

1. Gulli & Signorini
2. Azad & Abhishek
3. Ferrara et al.

مهم‌ترین اهداف آن‌که خوشه‌بندی اسناد می‌باشد متمایل گشت (ساندیا و همکاران[1]، ۲۰۱۶). روش‌های خوشه‌بندی زیادی تابه‌حال ارائه شده است که اکثراً برای خوشه‌بندی اسناد متنی بکار گرفته می‌شوند. در اغلب موارد برای خوشه‌بندی اسناد Html نیز از همان الگوریتم‌های خوشه‌بندی اسناد متنی استفاده می‌شود؛ اما سؤال مهم در اینجا این است که "کدامیک از الگوریتم‌های خوشه‌بندی برای خوشه‌بندی اسناد Html مناسب‌تر است؟". برای پاسخگویی به این سؤال بایستی از نیازمندی‌های ویژه خوشه‌بندی اسناد Html آگاهی بیشتری داشته باشیم.

- اسناد غالباً در حدود ۲۰۰-۱۰۰۰ کلمه منحصربه‌فرد دارند و این تعداد کلمه نشان‌دهنده ابعاد بالای ماتریس ایجادشده از مجموعه اسناد است. برای بالا بردن سرعت خوشه‌بندی و متناسب‌سازی آن با حجم بالای اسناد کنونی، روش خوشه‌بندی کننده بایستی استراتژی مناسبی برای کاهش ابعاد سند بدون کم کردن دقت خوشه‌بندی داشته باشد.
- ازآنجایی‌که اسناد فراوانی با عناوین چندگانه موجود است. روش خوشه‌بندی کننده بایستی امکان همپوشانی[2] مناسبی بین خوشه‌ها را داشته باشد.
- عنوان مناسب برای هر خوشه از دیگر الزامات برای داشتن نتایج مناسب خوشه‌بندی است لذا الگوریتم خوشه‌بندی بایستی خلاصه‌ای مربوط، مناسب و همچنین معنی‌دار برای هر خوشه ایجاد کند.
- الگوریتم بایستی توانایی کشف تعداد بهینه خوشه‌ها را در خود داشته باشد و نیاز به واردکردن تعداد خوشه‌ها توسط کاربر نباشد.

سیستم پیشنهادی کلیه نیازمندی‌های بالا را تا حد قابل قبولی پاسخ می‌دهد.

در این مقاله، به مسئله خوشه‌بندی اسناد، سیستم‌های موجود استخراج اطلاعات از اسناد Html و مشکلات پیش رو در خوشه‌بندی اسناد می‌پردازیم. در ادامه سیستم پیشنهادی برای خوشه‌بندی ارائه شده و مشخصات مجموعه اسناد انتخابی برای تست آن مشخص می‌گردد. در انتها معیارهای ارزیابی الگوریتم‌های خوشه‌بندی از آزمایش‌های الگوریتم و مقایسه آن با الگوریتم‌های موجود استخراج خواهد شد.

1. Sandhya et al.
2. Overlap

پیشینه پژوهش

خوشه‌بندی اسناد و الگوریتم‌های آن

خوشه‌بندی که یکی از اهداف داده‌کاوی می‌باشد، به فرآیند تقسیم مجموعه‌ای از داده‌ها (یا اشیا) به زیر کلاس‌هایی با مفهوم خوشه اطلاق می‌شود. به‌این‌ترتیب یک خوشه، یک سری داده‌های مشابه می‌باشد که همانند یک گروه واحد رفتار می‌کنند. لازم به ذکر است خوشه‌بندی همان کلاس‌بندی است، با این تفاوت که در آن کلاس‌ها از پیش تعریف‌شده و معین نمی‌باشند و عمل گروه‌بندی داده‌ها بدون نظارت[1] انجام می‌گیرد. تعیین میزان تشابه خوشه‌ها از طریق تابع تعیین فاصله بین بردارهای واژه‌های متناظر هر سند محاسبه می‌گردد. دو دسته‌ی کلی از الگوریتم‌های خوشه‌بندی، سلسله‌مراتبی و تفکیکی می‌باشند (اشتاینبک و همکاران[2]، ۲۰۰۰).

روش تفکیکی[3]

روش تفکیکی یکی از انواع الگوریتم‌های خوشه‌بندی است. رایج‌ترین الگوریتم‌های خوشه‌بندی اسناد مانند: K_Means و BuckShot در این دسته قرار می‌گیرند. این الگوریتم‌ها داده‌ها را به چندین زیرمجموعه تقسیم می‌کنند. به علت این‌که چک کردن همه‌ی زیرمجموعه‌های ممکن، امکان‌پذیر نیست. تابع‌های مکاشفه‌ای حریصانه‌ی خاصی به کار گرفته می‌شوند. در این الگوریتم‌ها به‌صورت تکراری نقاط بین k خوشه جابجا می‌شوند تا درنهایت به بهترین خوشه ممکن نسبت داده شوند. نقش اصلی را در این متدها روش‌های آماری و احتمالی دارند، به همین دلیل خوشه‌های تولیدشده دارای قابلیت تفسیرپذیری بالایی بوده و بسیار موردقبول می‌باشند (نا و همکاران[4]، ۲۰۱۰).

1. unsupervised
2. Steinbach et al.
3. Partitioning method
4. Na et al.

روش سلسله‌مراتبی[1]

الگوریتم‌های سلسله مراتبی به دو دسته تقسیم می‌شوند. الگوریتم‌های تجمیعی AHC[2] (پایین به بالا) و تقسیمی DHC[3] (بالا به پایین). در خوشه‌بندی تجمیعی، کار با خوشه‌هایی با یک داده شروع می‌شود (تعداد خوشه‌ها در ابتدا به اندازه‌ی تعداد داده‌های موجود می‌باشد). در هر مرحله دو یا چند خوشه‌ی مناسب با هم ترکیب‌شده و خوشه‌ی جدیدی را به وجود می‌آورند. در خوشه‌بندی تقسیمی عمل خوشه‌بندی با یک خوشه شروع می‌شود. این خوشه به‌صورت بازگشتی به دو یا چند خوشه تقسیم می‌گردد و به همین ترتیب عمل خوشه‌بندی ادامه پیدا می‌کند. برای هر دو نوع از الگوریتم‌های بالا ما نیاز به یک شرط پایانی داریم این شرط اغلب رسیدن به k خوشه می‌باشد. از مشهورترین این الگوریتم‌ها الگوریتم SingelPass است.

در حالت کلی در میان انواع مختلف الگوریتم‌های خوشه‌بندی، الگوریتم‌های تفکیکی سریع‌تر از الگوریتم‌های سلسله مراتبی هستند زیرا روش‌های سلسله مراتبی به علت اینکه بایستی تقریباً تمام روابط بین داده‌ها را در مجموعه داده موردنظر تجزیه‌وتحلیل و محاسبه نمایند لذا هزینه‌بری بیشتری ازنظر زمانی و حافظه دارند. ازجمله نقایص دو روش فوق موارد زیر را می‌توان نام برد (گامیر و پاتیل، ۲۰۱۵):

- همپوشانی در خوشه‌ها وجود ندارد و یک توصیف مناسب برای خوشه‌ها تولید نمی‌شود.
- تعیین تعداد خوشه‌ها با کاربر سیستم است و این الگوریتم‌ها برای کاهش ابعاد روشی را ارائه نمی‌دهند.
- توالی کلمات که در رساندن معنا بسیار مهم می‌باشند در این دو نوع الگوریتم در نظر گرفته نمی‌شود و با هر سند مانند مجموعه‌ای از کلمات برخورد می‌نمایند.

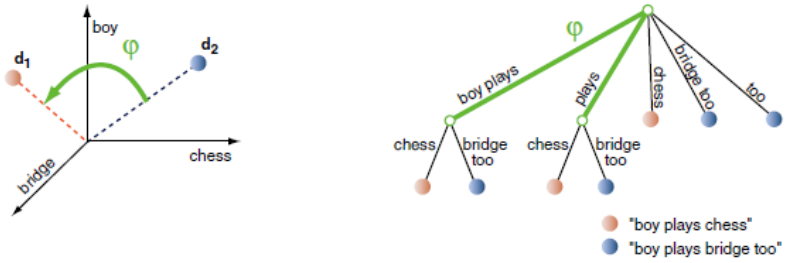
1. Hierarchical method
2. Agglomerative Hierarchical Clustering
3. Division Hierarchical Clustering

روش Suffix Tree

ایده استفاده از توالی کلمات (عبارات[1]) اولین بار در الگوریتمی به نام Stc ظهور پیدا کرد (ژوانگ و چن[2]، ۲۰۱۵). الگوریتم Stc برای خوشه‌بندی نتایج بازگشتی از موتور جستجو به کار می‌رود. در این الگوریتم به جای اینکه به اسناد همانند مجموعه‌ای از کلمات برخورد شود، هر سند به‌عنوان یک رشته در نظر گرفته می‌شود. مقایسه بین رشته‌ها به الگوریتم این امکان را می دهد که رابطه بین کلمات را از یک سند نسبت به سند دیگر به‌طور تقریبی محاسبه نماید. با این روش اسناد با توجه به عبارات مشترک در گروه‌های مشخصی خوشه‌بندی می‌شوند. ازآنجایی که این الگوریتم تنها برای خوشه‌بندی Snippet ها کارا است در مواجهه با اسناد متنی با حجم بالاتر، هزینه بری بالایی از نظر حافظه دارد.

مقایسه روش‌های مدل کردن اسناد

دو روش اصلی در مدل کردن سند وجود دارد. در شکل (۱) ساختار این دو مدل با هم مقایسه گردیده است. شکل سمت چپ نشان دهنده مدل برداری سند است. بردار سند سه کلمه معنی‌دار متمایز دارد که عبارت‌اند از "boy"، "chess" و "bridge" و هرکدام یک بعد برای بردار سند را تشکیل می‌دهند و تابع φ میزان تشابه این دو سند را محاسبه می‌نماید (زاویه دو بردار d1,d2) شکل سمت راست نشان‌دهنده ساختار suffix Treeبرای دو سند "boy plays chess" و "boy plays bridge too" می‌باشد و تابع φ نشان‌دهنده میزان همپوشانی این دو سند می‌باشد (با رنگ سبز مشخص می‌باشد).



شکل ۱. ساختار مدل‌های سند

1. phrases
2. Zhuang & Chen

سیستم‌های استخراج اطلاعات از اسناد HTML

در سال‌های اخیر تلاش‌های زیادی در زمینه استخراج اطلاعات از وب جهانی انجام‌گرفته است. سیستم استخراج اطلاعات به‌صورت سیستمی که جهت " شناسایی خودکار مجموعه از پیش تعریف‌شده‌ای از آیتم‌های مرتبط " استفاده می‌گردد تعریف می‌شود (اسپنگلر و گالیناری[1]، ۲۰۱۰). یک سیستم استخراج اطلاعات اسناد، مجموعه‌ای از اسناد را دریافت کرده و اطلاعات موردعلاقه کاربر سیستم را تا حدی که امکان‌پذیر است به‌صورت اتوماتیک از میان اسناد استخراج می‌نماید و به‌صورت کارایی در اختیار کاربر قرار می‌دهد. ازآنجایی‌که داده‌های باارزش زیادی در وب جهانی در قالب اسناد Html وجود دارد، تاکنون تلاش زیادی درباره استخراج اطلاعات از این اسناد انجام‌شده است (تار و نایانات[2]، ۲۰۱۱). اکثر سیستم‌های استخراج اطلاعات با این موضوع به‌مانند یک مسئله کلاس‌بندی برخورد کرده‌اند. در ادامه به برخی از مهم‌ترین این سیستم‌ها اشاره می‌گردد:

باتلر و همکاران (باتلرو همکاران[3]، ۲۰۰۱) برای استخراج اطلاعات جستجو شده کاربر در اسناد Html, Xml مدلی به نام "درخت تگ " برای هر سند ایجاد می‌کنند. هدف این مدل خلاصه‌سازی و کلاس‌بندی اطلاعات موجود بر اساس ارتباط آن با موضوع مورد جستجو می‌باشد. در این درخت گره‌های داخلی شامل تگ‌های Html,Xml و متن‌های این اسناد گره‌های انتهایی درخت (برگ‌ها) را تشکیل می‌دهند. با استفاده از یک الگوریتم اکتشافی تمامی اطلاعات به‌جز اطلاعات مرتبط با موضوع مورد جستجو از درخت حذف می‌گردد و درنهایت این درخت خلاصه‌شده به کاربر ارائه می‌شود.

کرسنزی و همکاران (کرسنزی و همکاران[4]، ۲۰۰۴) سیستمی برای استخراج اطلاعات از وب‌سایت‌های بزرگ ایجاد کردند که آن را "DataGarbber" نامیدند. این سیستم بعد از بررسی دقیق وب‌سایت مدلی برای آن استنتاج می‌نماید. در این مدل هر وب‌سایت به مانند گرافی در نظر گرفته می‌شود که گره‌های گراف را صفحات و یال‌های آن را لینک‌های بین صفحات تشکیل می‌دهند. با تحلیل لینک‌های هر صفحه میزان همبستگی صفحات و درنتیجه

1. Spengler & Gallinari
2. Tar & Nyunt
3. Buttler et al.
4. Crescenzi et al.

شباهت این صفحات به هم تشخیص داده می‌شود و از این طریق امکان کلاس‌بندی صفحات مشابه به وجود می‌آید.

امبلی و همکاران (امبلی و همکاران[1]، ۲۰۰۵) سیستمی را جهت استخراج اطلاعات از جداول داخل اسناد Html پیشنهاد کردند که استخراج اطلاعات را به‌صورت مرحله‌به‌مرحله انجام می‌دهد. در مرحله اول با توجه به سطرها و ستون‌های هر جدول و محتوی آن یک مدل مفهومی برای جدول در نظر گرفته می‌شود. در مرحله دوم مقادیر و صفات تعیین‌شده با توجه به محتوی جدول به مدل مفهومی اولیه اختصاص می یابد. در مرحله سوم صفات و مقادیر موجود در مدل بررسی و تنظیم می‌شود. در مرحله چهارم مدل تجزیه و تحلیل شده و اطلاعات اصلی استخراج می گردد. در نهایتا مراحل این استخراج اطلاعات از مبدا تا رسیدن به نتیجه ترسیم می‌شود.

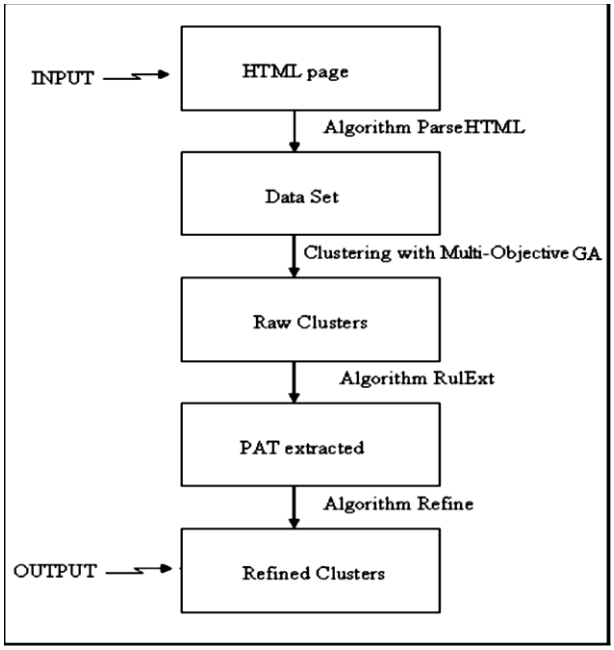
نگرش مرتبط دیگر در استخراج اطلاعات از اسناد Html در مقاله ای تحت عنوان "RoadRunner " در (فررا و همکاران، ۲۰۱۴) ارائه گردید. در این روش استخراج اطلاعات از وب سایتهای حاوی داده های فشرده که مستقیما رکوردهای داخل پایگاه داده را نمایش می دهد، مورد نظر می باشد. با دریافت حداقل ۲ سند از اینگونه وب‌سایت‌ها ساختار و بدنه اصلی این اسناد تشخیص داده می‌شود و امکان استخراج اطلاعات از این صفحات فراهم می گردد. مزیت این روش توانایی استخراج اطلاعات از صفحات دارای تگ‌های تودرتو می‌باشد. از معایب این روش نیاز به حداقل ۲ سند برای شناسایی ساختار صفحات است و دیگر اینکه در وب‌سایت‌هایی که صفحات آن به شیوه‌های متنوع ایجادشده‌اند نتایج ضعیفی به همراه دارد.

لابسکی و همکاران (لابسکی و همکاران[2]، ۲۰۰۵) تلاشی در جهت ایجاد سیستمی برای استخراج اطلاعات از اسناد Html که به‌عنوان کاتالوگ برای تولیدات محسوب می‌شوند، انجام دادند. در این سیستم از مدل مخفی مارکوف۳ جهت ایجاد یک موتور جستجوی پویا که قابلیت جوابگویی به سؤالات کاربر را داشته باشد، استفاده‌شده است. این موتور جستجو با نام "Semantic Search Engine" قادر به ارائه اطلاعاتی مانند نام، مشخصات، مزایا،

1. Embley et al.
2. Labsky et al.
3. HMMS

قیمت و ... مربوط به کالای تولیدی است و این کار را با استخراج اطلاعات از کاتالوگ ارائه شده برای هر محصول انجام می‌دهد. از نقاط قوت آن توانایی استخراج متن و تصویر با هم می‌باشد.

در سال ۲۰۰۸ مقاله‌ای تحت عنوان خودکار سازی استخراج اطلاعات از اسناد Html ارائه گردید. اشرف و زیر (اشرف و زیر[1]، ۲۰۰۸) سیستمی پیشنهادی به نام ClusTex را جهت استخراج اطلاعات از اسناد Html ارائه دادند. مهم‌ترین نکته‌ای که تفاوت این نگرش را با سایر روش‌های استخراج نمایان می‌ساخت، استفاده از تکنیک‌های خوشه‌بندی برای استخراج اطلاعات است. تکنیک‌های خوشه‌بندی جزء فنون یادگیری بدون نظارت هستند و به‌عبارت‌دیگر نیاز به وجود کاربر جهت ایجاد داده‌های آموزشی و همچنین نیاز به بازخوردهای کاربر در حین مراحل مختلف که در روش‌های کلاس‌بندی معمول می‌باشد در این تکنیک وجود ندارد. مراحل مختلف پردازش در این سیستم در شکل (۲) نشان داده‌شده است.



شکل ۲. مراحل مختلف پردازش در سیستم ClusTex

1. Ashraf & Zyer

گوپتا و گارگ[1] (۲۰۱۶) الگوریتمی به نام "شاخصه‌های وزن‌دهی شده K_means" ارائه دادند که در آن مجموعه داده‌های استاندارد را پس از پیش پردازش با جداسازی کلمات موجود در سند و حذف عبارات ایستا برای خوشه‌بندی آماده‌سازی می‌نمایند. در مرحله بعد برای کاهش ابعاد اسناد بر اساس یک حد آستانه[2] مشخص واژه‌هایی که از تعداد حد آستانه بیشتر هستند به عنوان ورودی مرحله بعد استفاده می‌شوند. در ادامه برای استخراج مشخصه‌های[3] هر سند از معیار TF/IDF[4] استفاده شده‌است با این فرض کلمات پرتکرار اهمیت بیشتری دارند. در مرحله بعد مشخصه‌های هر سند در قالب برداری استخراج می‌گردد؛ مجموع بردارها تشکیل یک ماتریس به نام VSM را می‌دهند که براساس نرخ مکسوبه[5] تشکیل شده است. پس از برای کاهش ابعاد ماتریس ایجاد شده از نسبت کسب شده هر یک از اسناد استفاده می‌شود. به عبارت دیگر اگر این نسبت بیشتر از حد آستانه مشخص شده باشد از ماتریس خلاصه شده VSM برای خوشه‌بندی استفاده خواهدشد.

مشکلات موجود در خوشه‌بندی اسناد HTML

با توجه به پیشینه خوشه‌بندی اسناد که در بخش ۲ ارائه گردید. می‌توان مشکلاتی که در خوشه‌بندی اسناد Html با آن روبرو هستیم را به‌صورت زیر جمع‌بندی نمود:

- اسناد Html مانند دیگر متون نوشته‌شده توسط انسان شامل توالی معناداری از کلمات می‌باشند؛ اما اکثر روش‌های ارائه‌شده تاکنون از مدل برداری استفاده می‌نمایند؛ که در خوشه‌بندی توالی کلمات را مدنظر قرار نمی‌دهد.
- ابعاد بسیار بالای این‌گونه اسناد از مهم‌ترین چالش‌های پیش رو می‌باشد. به‌عنوان مثال در وب‌سایت‌های گروه‌های خبری یا مقالات موجود در آن‌ها، دریک سند ممکن است بیش از ۱۰۰۰ کلمه منحصربه‌فرد موجود باشد. برای بالا بردن کارایی پردازش این اسناد بایستی الگوریتم‌های پیشنهادی راهی مناسب جهت کاهش ابعاد

1. Gupta & Garg
2. Threshold
3. Attribute
4. Term Frequency/ Inversed Document Frequency
5. Gain Ratio

اسناد ارائه نمایند. در حالی که الگوریتم‌های کنونی روش مناسبی برای این کار ارائه نداده‌اند.

- امکان همپوشانی اسناد در خوشه‌ها بایستی وجود داشته باشد. به‌عبارت‌دیگر بسیاری از اسناد دارای عناوین چندگانه هستند به‌عنوان‌مثال در یک سایت خبری همان‌طور که اخباری در رابطه با سلامت موجود است درباره جنگ نیز ممکن است وجود داشته باشد یا مقالاتی با موضوع نقش IT در فرهنگ اجتماعی می‌تواند از این‌گونه عناوین چندگانه به شمار آید.
- الگوریتم پیشنهادی بایستی توانایی ایجاد برچسب‌های خلاصه و مفیدی که راهنمای کاربر در فهمیدن محتوی خوشه باشد را داشته باشد به عبارتی دیگر برچسب هر خوشه توصیف دقیق و خلاصه‌ای از محتوی آن خوشه می‌باشد. در صورتی که در اغلب این الگوریتم‌ها عنوان مناسبی به خوشه‌ها داده نمی‌شود.
- تعداد خوشه‌های مناسب برای هر مجموعه از اسناد بایستی توسط خود الگوریتم و با توجه به خصیصه‌های موجود در آن مجموعه تعیین شود. درحالی‌که در الگوریتم‌های تفکیکی و سلسله مراتبی برای تعیین شرط پایان نیاز به دانستن تعداد خوشه‌ها قبل از اجرای الگوریتم می‌باشد.

در الگوریتم Ks_Stc برای خوشه‌بندی اسناد Html موارد زیر رعایت گردیده است:

- مدل‌های برداری اسناد توالی کلمات را در سند در نظر نمی‌گیرند. درحالی‌که در متن توالی کلمات نقش بسیار بارزی را در رساندن معنا به کاربر بازی می‌کند و عوض شدن موقعیت کلمه در جمله می‌تواند معنای جمله را به‌کلی تحت تأثیر خود قرار دهد. به‌عنوان‌مثال عبارتی " Association Rule " یک مفهوم کاملاً مشخص درداده کاوی است، حال اگر در جمله‌ای این دو کلمه جابجا شوند مثلاً " ... The rule of association" که معنی آن کاملاً متفاوت است. لذا الگوریتم پیشنهادی ما از مدل ST برای مدل کردن اسناد استفاده نموده است. در این مدل توالی کلمات نیز در مدل کردن سند تأثیرگذار است.

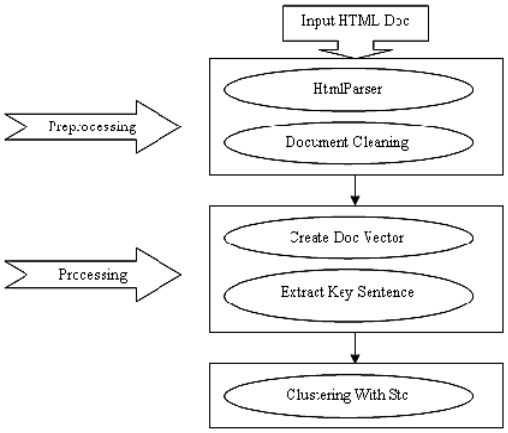
کاهش ابعاد سند در الگوریتم پیشنهادی بر سه محور استوار است که عبارت‌اند از:

- استفاده از تگ‌های Html برای خلاصه‌سازی تا خلاصه‌سازی کمترین تأثیر را بر قسمت‌های مهم هر سند مانند عناوین داشته باشد.
- استفاده از اقلام تکراری جهت خلاصه‌سازی بهتر ازلحاظ آماری تا حد امکان بهترین خلاصه‌سازی از محتوای اسناد html به عمل آید و نتایج حاصله قابلیت تفسیرپذیری مناسب‌تری داشته باشند.
- ازآنجایی‌که تنها فعل‌ها و اسم‌ها حاوی اطلاعات اصلی می‌باشند. با استفاده از یک لغتنامه ۴۴۰۰۰ کلمه‌ای که تنها شامل اسم‌ها و فعل‌های لاتین می‌باشد، کلیه کلمات ایستا را که بار اطلاعاتی ندارند از سند حذف می‌نماید.

روش‌شناسی پژوهش

الگوریتم پیشنهادی برای خوشه‌بندی اسناد HTML

سیستم پیشنهادی از سه بخش اصلی تشکیل یافته است در شکل (۳) شمای کلی سیستم پیشنهادی ارائه‌شده است.



شکل ۳. شمای کلی سیستم Ks_Stc

پیش‌پردازش

این بخش شامل مراحل زیر است:

جداکننده تگ‌های Html:

الگوریتمی جهت استخراج تگ‌های موردنظر ایجادشده است و شبه کد الگوریتم در شکل (۴) ارائه می‌گردد. تگ‌های موردنظر در آرایه‌ای به نام (DesiredTag) ذخیره می‌گردد. این آرایه شامل تگ شروع، پایان و ارزش هر تگ می‌باشد. در اولین مرحله تمام تگ‌های موجود در سند یک‌به‌یک در آرایه‌ای به نام (Tok) ذخیره می‌گردد و متن‌های نظیر این تگ‌ها (مابین دو تگ ابتدایی و پایانی) در آرایه‌ای دیگر به نام (Tokens) ذخیره می‌شود. در مرحله بعدی آرایه تگ‌ها (Tok) با آرایه (DesiredTag) مقایسه می‌شود؛ و در صورت مطابقت داشتن تگ استخراج‌شده از متن با تگ‌های دلخواه، تمام متن مربوط به تگ از آرایه متناظر (Tokens) استخراج می‌گردد؛ و ارزش کلیه تگ‌های مابین آن نیز به همراه متن در رشته خروجی الگوریتم ذخیره می‌شود.

شکل ٤. شبه کد الگوریتم HtmlParser

```
Algorithm: HtmlParser
Input: Html File
Begin
    While input_file is not empty
        Read Character ch from input_file
        Save into char_array
        num_char=location of last character in char_array
    For i=0 to num_char
        Read ch at char_array[i]
        If (ch='<') /*this must be start tag
            Start making a string for Html tag
        Else if (ch='>') /* this must be the end of a tag
            Complete and save tag in array Tok
            num_tok=location of last token in Tok
            Make an empty string to store text token.
        Else
            Save ch in tag or text string.
            If ch at char_array [i+1] is '<'
                Save text string in array Tokens
                num_tok=location of last token in Tok

/* arrayList desiredTag have Start Tag & End Tag & Weight of This Tag
/*OutputStr is a empty string

    For j=0 to num_tok
        If Tok[j] at arrayList desiredTag
            OutputStr+=Weight
            While Tok[j] is not EndTag
            OutputStr+=Tokens[j]
    Output string OutputStr
End HtmlParser
```

پاک‌سازی اسناد

در این مرحله بر هر سند (رشته خروجی از مرحله اول) موارد ذیل اعمال می‌گردد:

- تغییر افعال به ریشه (مصدر) با حذف پیشوندها پسوندها، تغییر حالت جمع به مفرد

ازآنجایی‌که تنها فعل‌ها اسم‌ها حاوی اطلاعات اصلی می‌باشند (شارقی و همکاران[1]، ۲۰۱۵). با استفاده از یک لغتنامه ۴۴۰۰۰ کلمه‌ای که تنها شامل اسم‌ها و فعل‌های لاتین می‌باشد به پاک‌سازی اسناد دریافتی می‌پردازیم. نحوه عمل بدین‌صورت است که کل سند خلاصه‌شده از مرحله اول را بر اساس Space جداسازی می‌نماییم و هر کلمه به‌دست‌آمده را با این لغتنامه تطبیق می‌دهیم. در صورت وجود داشتن کلمه در لغتنامه عدد متناظر هر کلمه (کد کلمه) را در فایلی ذخیره می‌نماییم. با این کار علاوه بر کد نمودن کلمات تمام کلمات ایستا را که بار اطلاعاتی برای ما ندارند مانند نشانه‌ها، اعداد، بالت‌ها و... حذف نموده‌ایم البته غیر از نقطه‌ها. (در این لغتنامه به جهت نیاز به نقطه جهت تشخیص جملات کد کلمه صفر به نشانه نقطه اختصاص‌یافته است.)

پردازش

ایجاد بردار وزن‌دار و تعیین کلمات کلیدی

در این مرحله برای هر سند پاک‌سازی‌شده بردار وزن‌داری ایجاد می‌شود. لازم به توضیح است که هر سند پاک‌سازی‌شده شامل توالی‌هایی از کد کلمات است و به هر توالی ارزشی که ناشی از تگ یا تگ‌های مربوطه می‌باشد اختصاص‌یافته است. در اینجا از یک تابع Hash مناسب برای شمارش کد هر کلمه و محاسبه مجموع ارزش‌های آن استفاده‌شده است تا سرعت اجرای این کار را بالا ببرد. درنهایت کد کلمات را بر اساس ارزش محاسبه‌شده برای آن‌ها، مرتب می‌نماییم تا بردار وزن‌دار را ایجاد نماید. به‌صورت خلاصه هر بردار وزن‌دار شامل تمام کلمات موجود در سند پاک‌سازی‌شده همراه با وزن (ارزش) آن کلمه می‌باشد که بر اساس ارزش مرتب گردیده است. ارزش هر کلمه شامل مجموع ارزش اکتسابی کلمه از تگ‌هایی که در آن وجود داشته است.

1. Shareghi et al.

استخراج جملات کلیدی

قبل از توضیح این مرحله دو تعریف مهم ارائه می‌گردد:

کلمات کلیدی: ۲۰ کلمه اول بردار وزن‌دار (کلمات با بالاترین وزن) کلمات کلیدی فرض شده‌اند.

جملات کلیدی: جمله‌هایی که حداقل شامل ۲ کلمه کلیدی است (MinimumSupport=2) و حداکثر طول هر جمله کلیدی شامل ۲۵ کلمه می‌باشد.

در این مرحله سند پاک‌سازی‌شده بر اساس کد کلمه صفر (نقطه) تفکیک می‌شود. از طرفی در صورت بزرگ‌تر بودن طول جملات تفکیک‌شده از ۲۵ کلمه، جمله بر اساس ۲۵ کلمه به چند جمله تفکیک می‌گردد. در این حالت مجموعه جملات شناسایی‌شده را به‌عنوان ورودی به الگوریتمی که باکمی تغییر از الگوریتم Apriori گرفته‌شده است داده می‌شود. کار این الگوریتم تعیین جملاتی از این مجموعه جملات است که شرط کلیدی بودن (حداقل شامل دو کلمه کلیدی) و به عبارتی Minimum Support=2 را داشته باشند. با آزمایش‌های به‌عمل‌آمده متوسط تعداد جملات استخراج‌شده برای یک سند با ۳۵۰ کلمه متمایز، ۱۷ جمله کلیدی می‌باشد؛ که این جملات خلاصه کاملاً مناسبی برای هر سند ایجاد می‌نمایند.

خوشه‌بندی با استفاده از جملات کلیدی استخراج‌شده

در این مرحله که بعد از تعیین جملات کلیدی تمام مجموعه اسناد انجام می‌گیرد. ابتدا از جملات کلیدی به‌دست‌آمده درخت پسوندی ایجاد می‌گردد. سپس مرحله شناسایی خوشه‌های پایه انجام می‌گیرد و با ترکیب خوشه‌های پایه خوشه‌های نهایی ایجاد می‌شوند و نتایج به شکل مناسبی در اختیار کاربر گذاشته می‌شود. (کلیه موارد اشاره‌شده از الگوریتم Stc برداشت‌شده است).

مجموعه اسناد فراهم‌شده برای آزمایش‌ها

برای آزمایش الگوریتم پیشنهادی ۱۰ مجموعه با مشخصاتی که در جدول (۱) آمده، تهیه شده است. این مجموعه اسناد از مجموعه اسناد استاندارد در (وبسایت پدال، ۲۰۱۳) می‌باشد

که برای ارزیابی الگوریتم‌های خوشه‌بندی به کار می‌رود کل این مجموعه اسناد شامل ۱۱۰۰۰ سند در ۱۱ کلاس مختلف و با حجمی بالغ بر ۱٫۳ GB می‌باشد و این اسناد توسط عامل انسانی بر اساس محتوی آن دسته‌بندی‌شده است. ازآنجایی‌که نیاز به چندین مجموعه سند احساس می‌گردد، این مجموعه‌ها با دقت کامل از مجموعه اصلی برداشت‌شده‌اند.

جدول ۱. مشخصات مجموعه‌های اسناد برای تست الگوریتم

شماره مجموعه داده ها	تعداد اسناد	تعداد گروه ها	میانگین اندازه گروه ها	میانگین طول اسناد	مجموع اندازه (مگابایت)
۱	۳۴۰	۷	۵۳	٦۹	۲۳/۵
۲	۸۰۷	۹	۹۸	۱٤٤	۱۱۶
۳	۷۹۷	۱۱	۷۳	۵۲	۴۱/۷
۴	۱۷۶۱	۱۰	۱۶۵	۸٤	۱۴۸/۲
۵	۴۴۴۳	۸	۵۵۵	٦٤	۲۸۴
۶	۲۶۲	۴	۹۳	۷٦	۱۱۹/۵
۷	۳۵۵	۴	۱۰۸	۱۰۵	۵۰۱/۴
۸	۱۷۸	۳	۶۵	۱۱۵	۲۰۸
۹	۶۰۶۴	۱۰	۶۰۵	۵۲	۳۱۶/۸
۱۰	۲۰۰	۴	۵۴	۷۳٦	۱۴۸/۲

روش‌های ارزیابی استاندارد برای الگوریتم‌های خوشه‌بندی

دو معیار استاندارد برای ارزیابی صحت الگوریتم‌های خوشه‌بندی با عنوان F_Measure و Purity وجود دارد.F_Measure یک ترکیب هارمونیک از دو فاکتور P(i, j)[1] و R(i, j)[2] است که در بازیابی اطلاعات استفاده می‌شود (اشتاینبک و همکاران، ۲۰۰۰).

$$P(i, j) = \frac{n_{ij}}{n_j} \quad (۱)$$

1. precision
2. recall

(۲)
$$R(i, j) = \frac{n_{ij}}{n_i}$$

که در آن ni تعداد اعضای کلاس i، nj تعداد اعضای خوشه j، nij تعداد اعضای کلاس i در خوشه j می‌باشند. F(i, j)[1] به‌صورت زیر برای هر خوشه تعریف می‌شود:

(۳)
$$F = \sum_i \frac{n_i}{n} \max_j (F(i, j))$$

و برای کل نتایج خوشه‌بندی داریم:

(۴)
$$F(i, j) = \frac{2 * P(i, j) * R(i, j)}{P(i, j) + R(i, j)}$$

که در آن n نشان‌دهنده کل اسناد در مجموعه داده است. به‌طور خلاصه هرچه F بزرگ‌تر باشد نشان‌دهنده بالاتر بودن صحت عمل الگوریتم است (اشتاینبک و همکاران، ۲۰۰۰). معیار Purity برای ارزیابی میزان خلوص خوشه‌های تولیدی الگوریتم می‌باشد و به‌صورت زیر تعریف می‌شود

(۵)
$$Purity(j) = \frac{1}{n_j} \max(n_{ij})$$

یافته‌های پژوهش

اعتبارسنجی الگوریتم

طبق پژوهش انجام شده توسط ژونجی و همکاران (۲۰۰۹) F_Measure معیار مناسبی برای اعتبارسنجی الگوریتم‌های خوشه‌بندی است. معیار F_Measure متشکل از دو زیرمعیار دقت و بازیابی اطلاعات اطلاعات می‌باشد. دقت به معنای نرخ اطلاعات مرتبط به کل نتایج جست‌وجوی سند موردنظر و بازیابی به معنای نسبت اطلاعات مرتبط در نتایج جست‌وجوی سند به کل داده‌های مجموعه می‌باشد. به‌طور خلاصه هرچه F بزرگ‌تر باشد نشان‌دهنده بالاتر بودن صحت عمل و اعتبار الگوریتم است.

1. F_Measure

جدول ۲. مقایسه معیار F_Measure الگوریتم های مختلف

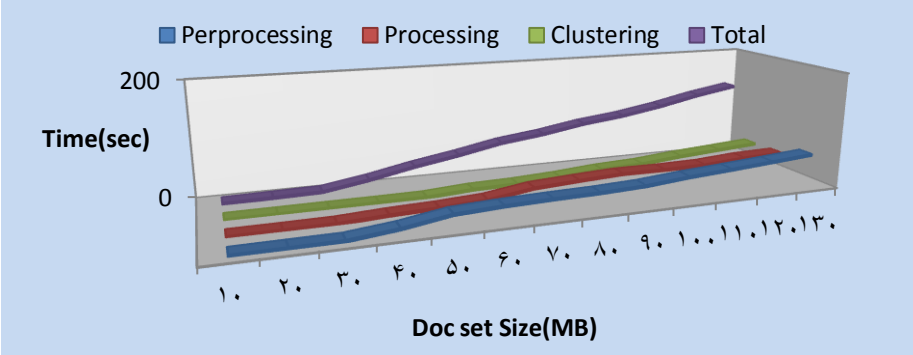
	DS 1	DS 2	DS 3	DS 4	DS 5	DS 6	DS 7	DS 8	DS 9	DS1 0
K_means	۰/۲۹	۰/۶۱	۰/۴۹	۰/۳۵	۰/۲۲	۰/۴۲	۰/۲۱	۰/۴۸	۰/۳۶	۰/۵۱
Buckshot	۰/۱۳	۰/۵۰	۰/۵۸	۰/۳۲	۰/۱	۰/۰۹	۰/۳۷	۰/۳۹	۰/۳۱	۰/۱۴
Single_Pas s	۰/۲۲	۰/۳۸	۰/۵۹	۰/۱۹	۰/۳	۰/۵۱	۰/۴۰	۰/۵۵	۰/۳۱	۰/۶۰
KS_Stc	۰/۳۵	۰/۷۵	۰/۶۲	۰/۵۴	۰/۴۲	۰/۶۹	۰/۵۳	۰/۷۰	۰/۴۵	۰/۷۴

نتایج آزمایش‌های الگوریتم KS_Stc و مقایسه آن با سایر الگوریتم‌های موجود

برای انجام آزمایش‌ها نیاز به پیاده‌سازی الگوریتم‌های Kmeans (نا و همکاران، ۲۰۱۰)، BuckShot (گامیر و پاتیل[1]، ۲۰۱۵)، SingelPass (ژیائولین و همکاران[2]، ۲۰۱۳) و همچنین الگوریتم پیشنهادی Ks_Stc وجود داشت. لذا تمام الگوریتم‌های بالا در محیط C#.NET و کاملاً مطابق با استاندارد اصلی الگوریتم‌ها که از منابع معین‌شده به دست آمد پیاده‌سازی گردیده است. کلیه آزمایش‌های زیر بر روی pc با cpu AMD Athlon 1500+,60 GB HD,1024 MB Ram انجام‌شده است.

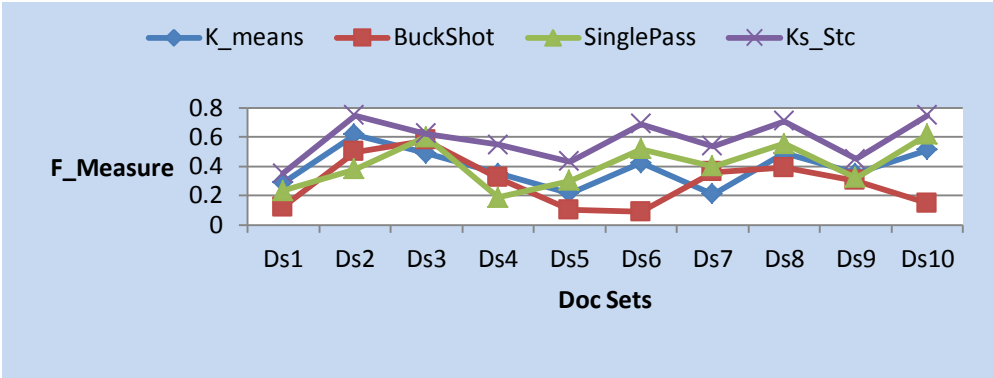
در اولین تست الگوریتم Ks_Stc با مجموعه‌های اسناد در حجم‌های متفاوت آزمایش گردید و زمان موردنیاز برای اجرای کل مراحل الگوریتم محاسبه شد. همان‌گونه که در شکل (۵) مشاهده می شود. زمان اجرای هرکدام از سه مرحله اجرای الگوریتم با افزایش حجم مجموعه اسناد به‌صورت خطی افزایش می‌یابد. بیشترین زمان را مرحله PerProcessing و کمترین زمان را بخش Clustering در کل مراحل اجرای الگوریتم به خود اختصاص داده است. بر اساس نتایج به‌دست‌آمده می‌توان خطی بودن میزان افزایش زمان اجرا را با افزایش حجم مجموعه اسناد نتیجه گرفت.

1. Gamare & Patil
2. Xiaolin et al.



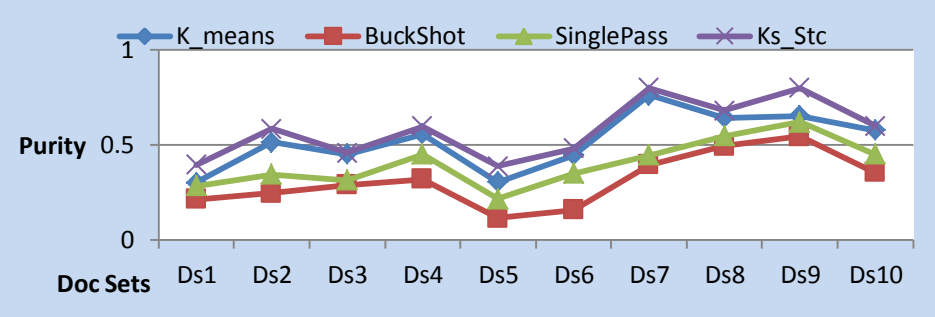
شکل ۵. زمان موردنیاز برای اجرای مراحل سه‌گانه الگوریتم

در تست دوم الگوریتم Ks_Stc ازلحاظ معیار F_Measure با سه الگوریتم متداول خوشه‌بندی اسناد مقایسه می‌گردد و نتایج در شکل (۶) ارائه‌شده است. همان‌گونه که اشاره شد معیار F_Measure صحت عملکرد الگوریتم را بیان می‌نماید. هرچه این معیار بزرگ‌تر باشد عمل خوشه‌بندی با صحت بیشتری انجام گرفته است. ازآنجایی که حداکثر میزان صحت (کاملاً بدون اشکال بودن) برابر ۱ می‌باشد و مسئله خوشه‌بندی جزء مسائل بهینه‌سازی است، مقایسه الگوریتم‌ها در میزان نزدیک‌تر بودن به حالت بهینه انجام می‌گیرد. نتایج به‌دست‌آمده به‌وضوح بهتر بودن عملکرد الگوریتم پیشنهادی ازلحاظ صحت را نسبت به سایر الگوریتم‌ها نشان می‌دهد.



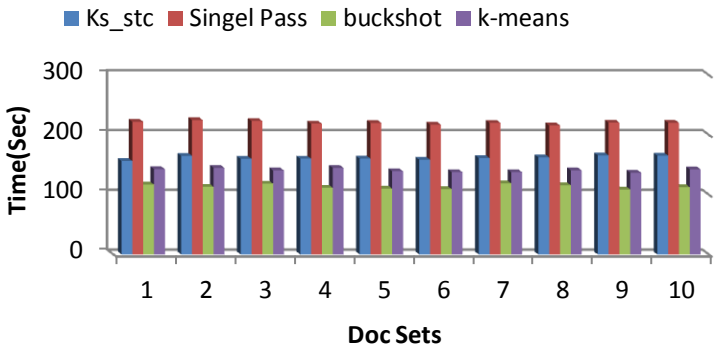
شکل ٦. مقایسه ۳ الگوریتم پرکاربرد با الگوریتم پیشنهادی ازنظر معیار F_Measure

در تست سوم الگوریتم KS_Stc ازلحاظ معیار Purity با سه الگوریتم متداول خوشه‌بندی اسناد مقایسه می گردد و نتایج در شکل (۷) ارائه‌شده است. همان‌گونه که اشاره شد معیار Purity درجه خلوص خوشه‌های ایجادشده را در الگوریتم بیان می‌نماید. هرچه این معیار بزرگ‌تر باشد عمل خوشه‌بندی با دقت بیشتری انجام‌گرفته و خوشه‌های ایجادشده باکیفیت‌تر می‌باشند. مقایسه الگوریتم‌ها در میزان نزدیک‌تر بودن به حالت ایده‌آل (Purity=1)، کاراتر بودن عملکرد الگوریتم پیشنهادی را نسبت به سایر الگوریتم‌ها نشان می‌دهد. دو الگوریتم BuckShot و SinglePass خوشه‌های ناخالص‌تری از دو الگوریتم دیگر ایجاد نموده‌اند. در ضمن الگوریتم پیشنهادی در هر ۱۰ مجموعه داده نتایجی بهتر از الگوریتم K-Means که استاندارد الگوریتم‌های خوشه‌بندی در این معیار می‌باشد ایجاد نموده است.



شکل ۷. مقایسه ۳ الگوریتم پرکاربرد با الگوریتم پیشنهادی ازنظر معیار Purity

در چهارمین آزمایش الگوریتم پیشنهادی با سه الگوریتم ازنظر زمان اجرا باهم مقایسه شده‌اند. مجموعه اسنادی شامل ۱۰۰۰ سند که هرکدام به‌طور متوسط دارای ۷۱۰ کلمه می‌باشند را برای این آزمایش انتخاب‌شده است و برای دستیابی به میانگینی قابل‌اعتماد ده مرتبه هر الگوریتم برای خوشه‌بندی این مجموعه اسناد استفاده گردید؛ و نتایج حاصله برای مقایسه بر روی نمودار آورده شده است. همان‌طور که در شکل (۸) و جدول (۲) دیده می‌شود سریع‌ترین الگوریتم خوشه‌بندی BuckShot و کندترین آن‌ها SingelPass می‌باشد. الگوریتم K_means سرعت میانگینی دارد و الگوریتم Ks_Stc باکمی کندتر بودن ازلحاظ سرعت به‌عنوان سومین الگوریتم مطرح می‌باشد.



شکل ۸. مقایسه ۳ الگوریتم پرکاربرد با الگوریتم پیشنهادی ازنظر زمان اجرا

جدول ۳. میانگین نتایج زمانه‌ای اجرای الگوریتم

الگوریتم	مقدار متوسط	رتبه (از کم به زیاد)	مقدار میانگین
k-means	۱۴۰/۶	۱۳۶/۷-۱۴۴/۸	۱۴۰/۷۵
Ks_Stc	۱۶۱/۴	۱۵۶/۲-۱۶۵/۶	۱۶۰/۹
Buckshot	۱۱۳/۵	۱۰۸/۵-۱۱۸/۹	۱۱۳/۷
SinglePass	۲۲۰/۲	۲۱۵/۸-۲۲۴/۷	۲۲۰/۲۵

سیستم پیشنهادی کلیه نیازمندی‌های خاص اسناد Html به شرح زیر را تا حد قابل قبولی پاسخ می‌دهد:

- در سیستم Ks_Stc روش جدیدی برای خلاصه‌سازی هر سند ارائه‌شده است؛ که با کاهش چشمگیری در ابعاد هر سند کمترین تأثیر را در دقت خوشه‌بندی و کیفیت خوشه‌های نهایی ایجاد می‌کند.

سیستم ارائه‌شده از تکنیک Stc برای خوشه‌بندی نهایی استفاده می‌کند. در این تکنیک به‌جای مدل برداری از ساختار داده‌ای به نام Suffix Tree استفاده می‌کند این ساختار مزیت‌های ذیل را در خوشه‌بندی ایجاد می‌کند (نا و همکاران، ۲۰۱۰).

- توجه به توالی کلمات در خوشه‌بندی که به ایجاد عنوان معنادار و مناسب برای هر خوشه می‌انجامد.

- همپوشانی مناسب را در خوشه‌های نهایی ایجاد می‌نماید.
- تعیین مناسب‌ترین تعداد خوشه‌ها به‌صورت خودکار با پردازش مجموعه اسناد در الگوریتم انجام می‌گیرد.

با مشاهده نتایج آزمایش‌ها می‌توان گفت که الگوریتم پیشنهادی ازلحاظ زمان اجرا نسبت به حجم مجموعه سند به‌صورت خطی عمل می‌نماید. هرچند نسبت به الگوریتم BuckShot,K_Means کندتر عمل می‌کند، اما کیفیت و درجه خلوص خوشه‌های تولیدشده آن بالاتر از این دو الگوریتم می‌باشد و نهایتاً نسبت به الگوریتم‌های متداول خوشه‌بندی متن، برای خوشه‌بندی اسناد Html کارایی بالاتری را دارد.

نتیجه‌گیری و پیشنهادها

در این مقاله روش جدیدی برای خوشه‌بندی اسناد Html بر اساس استخراج جملات کلیدی و با استفاده از الگوریتم‌های تلفیقی خوشه‌بندی اسناد ارائه گردید. سیستم ارائه‌شده و نتایج به‌دست‌آمده در آزمایش‌ها به‌وضوح کارایی و مؤثر بودن روش‌های تلفیقی را در خوشه‌بندی نشان داد. یکی از راه‌های ایجاد بهبود در نتایج به‌دست‌آمده از الگوریتم‌های خوشه‌بندی توجه بیشتر به روش‌های کاهش ابعاد و ارائه روش‌های نو و کاربردی در خلاصه‌سازی اسناد است. هرچه میزان دقت و کاهش ابعاد سند در این‌گونه روش‌ها مناسب‌تر باشند، نتایج نهایی باکیفیت و سرعت بهتری همراه خواهد بود. از طرفی می‌توان از الگوریتم‌های موجود در زمینه‌های دیگر داده‌کاوی و تلفیق آن‌ها با الگوریتم‌های خوشه‌بندی جهت افزایش کارایی این الگوریتم‌ها استفاده نمود ازجمله مهم‌ترین الگوریتمی که در این سیستم جهت تشخیص جملات کلیدی استفاده شد الگوریتم Apriori که در استخراج قوانین انجمنی کاربرد دارد (فیضی و همکاران[1]، ۱۳۹۳). استفاده از تابع Hash نیز از دیگر تکنیک‌هایی بود که به کارایی روش پیاده‌سازی شده کمک نمود در ادامه این تحقیق پیشنهاد می‌گردد، روش خلاصه‌سازی اسناد Html که در این مقاله ارائه گردید را با سایر الگوریتم‌ها مانند K_Means تلفیق نمود و میزان بهبود عملکرد آن‌ها را بررسی کرد.

1. Feyzi et al.

منابع

Anon., n.d. Html Document Set. [Online] Available at: http://www.pedal.rdg.ac.uk/banksearchdataset [Accessed 2013]

Ashraf, F. & Zyer, T. O., 2008. Employing Clustering Techniques for Automatic Information Extraction from HTML Documents. *IEEE Transactions on Syst.Man.Cyber,* 38(5), pp. 660-673.

Azad, H. K. & Abhishek, K., 2014. *Semantic-synaptic web mining: A novel model for improving the web mining. In Communication Systems and Network Technologies (CSNT),*. s.l., s.n., pp. 454-457.

Buttler, D., Liu, L. & Pu, C., 2001. *A fully automated object extraction system for the world wide web*. s.l., s.n., pp. 361-370.

Crescenzi, S. V., Mecca, G., Merialdo, P. & Missier, P., 2004. *An automatic data grabber for large Web sites.* s.l., s.n., pp. 1321-1324.

Embley, D. W., Tao, C. & Liddle, S. W., 2005. Automating the extraction of data from HTML tables with unknown structure. *Data Knowledge Engineering,* 54(1), pp. 3-28.

Ferrara, E., De Meo, P., Fiumara, G. & Baumgartner, R., 2014. Web data extraction, applications and techniques: A survey.. *Knowledge-Based Systems,* Volume 70, pp. 301-323.

Feyzi, K., Sabet Motlagh, M. & Abedini naeini, M., 1393. Using an integrated approach of QFD, FAHP, VIKOR to select the most suitable ERP system. *Journal of Management Studies Information Technology,* pp. 1-20.

Gamare, P. S. & Patil, G. A., 2015. *Efficient Clustering of Web Documents Using Hybrid Approach in Data Mining.* s.l., IEEE.

Gulli, A. & Signorini, A., 2005. *The indexable web is more than 11.5 bilion pages.* s.l., s.n., pp. 902-903.

Gupta, M. & Garg, K., 2016. Attribute Weighted K-means For Document Clustering. *International Research Journal of Engineering and Technology*, 3(6), pp. 1583-1590.

Junjie, W., Xiong, H. & Jian, C., 2009. Towards understanding hierarchical clustering: A data distribution perspective. *Neurocomputing*, pp. 2319-2330.

Labsky, M., Svatek, V., Praks, P. & Svab, O., 2005. *Information extraction from HTML product catalogues: Coupling quantitative and knowledge-based approaches.* Dagstuhl, Germany, s.n.

Na, S., Xumin, L. & Yong, G., 2010. *Research on k-means clustering algorithm: An improved k-means clustering algorithm. In Intelligent Information Technology and Security Informatics (IITSI).* s.l., 2010 IEEE Third International Symposium, pp. 63-67.

Sandhya, N., Govardhan, A. & Rameshchandra, G., 2016. Concept Based Text Document Clustering with Vector Suffix Tree Document Model. *International Journal of Computer Science and Information Security*, 14(7), p. 259.

Shareghi, E., Petri, M., Haffari, G. & Cohn, T., 2015. *Compact, Efficient and Unlimited Capacity: Language Modeling with Compressed Suffix Trees.* s.l., s.n., pp. 2409-2418.

Spengler, A. & Gallinari, P., 2010. *Document structure meets page layout: loopy random fields for web news content extraction.* s.l., s.n., pp. 150-160.

Steinbach, M., Karypis, G. & Kumar, V., 2000. *A comparison of document clustering techniques.* s.l.:s.n.

Tar, H. H. & Nyunt, T. S., 2011. Ontology-based concept weighting for text documents. *World Academy of Science, engineering and Technology*, Volume 57, pp. 249-253.

Xiaolin, Y., Xiao, Z., Nan, K. & Fengchao, Z., 2013. An improved Single-Pass clustering algorithm internet-oriented network topic detection. s.l., *Intelligent Control and Information Processing* (ICICIP), pp. 560-564.

Zhuang, Y. & Chen, Y., 2015. *Improving Suffix Tree Clustering Algorithm for Web Documents*. s.l., IEEE.